DEEP LEARNING BASED IMAGE SEGMENTATION FOR EXPANSIVE SOIL DESICCATION CRACK RECOGNITION AND QUANTIFICATION

LING HUI YEAN

MASTER OF ENGINEERING SCIENCE

LEE KONG CHIAN FACULTY OF ENGINEERING AND SCIENCE
UNIVERSITI TUNKU ABDUL RAHMAN
JANUARY 2025

DEEP LEARNING BASED IMAGE SEGMENTATION FOR EXPANSIVE SOIL DESICCATION CRACK RECOGNITION AND QUANTIFICATION

By

LING HUI YEAN

A dissertation submitted to the Department of Civil Engineering,
Lee Kong Chian Faculty of Engineering and Science,
Universiti Tunku Abdul Rahman,
in partial fulfillment of the requirements for the degree of
Master of Engineering Science
January 2025

COPYRIGHT STATEMENT

This dissertation is submitted in partial fulfilment of the requirements for the degree of Master of Engineering Science at Universiti Tunku Abdul Rahman (UTAR). This dissertation represents the work of the author, except where due acknowledgement has been made in the text. No part of this dissertation may be reproduced, stored, or transmitted in any form or by any means, whether electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the author or UTAR, in accordance with UTAR's Intellectual Property Policy.

ABSTRACT

DEEP LEARNING BASED IMAGE SEGMENTATION FOR EXPANSIVE SOIL DESICCATION CRACK RECOGNITION AND QUANTIFICATION

Ling Hui Yean

Expansive soils undergo significant volume changes due to moisture fluctuations, which lead to desiccation cracks formation that affect soil properties and engineering performance, compromising the safety of geo-structures. The analysis of these cracks was essential for mitigating their impact; however, traditional quantification methods were labour-intensive and imprecise, highlighting the need for more robust and automated techniques. This study investigated the feasibility and effectiveness of image-based techniques using advanced deep learning algorithms to quantify desiccation cracks in expansive soils. The objectives of the study included designing soil desiccation experiment setup for desiccation crack image acquisition, evaluating crack imaging analysis based on deep learning algorithms, and quantifying desiccation cracks through image processing techniques. Laboratory experiments were conducted using a custom-built image acquisition tool to capture crack images under simulated soil desiccation conditions. Crack images obtained were processed and annotated to produce a dataset of 820 images for the training and testing of deep learning models. Deep learning models, including U-Net, Res-UNet, and DeepLabv3+ with pre-trained backbones such as MobileNetV2, ResNet-18, ResNet-50, and

Xception, were trained and evaluated alongside a traditional Otsu's thresholding method as the baseline for crack detection and segmentation. The evaluation considered segmentation performance using evaluation metrics (precision, recall, F1 score, IoU), computational efficiency, and crack geometrical parameters quantification (surface crack ratio, crack width, crack length, and crack segment). Results demonstrated that DeepLabv3+ variants consistently outperformed other methods, with MobileNetV2 backbone offering the best balance of computational efficiency, segmentation accuracy, and robustness across case-wise performance conditions. Compared to traditional approaches, deep learning models, particularly with DeepLabv3+ variants, produced more reliable crack segmentation masks, thus enabling more accurate quantification of crack geometrical parameters, as demonstrated by lower error rates. This study validates the effectiveness of deep learning-based segmentation methods for automated soil crack recognition and quantification, contributing to engineering applications with improved methodologies for analysing desiccation behaviour in expansive soils.

Keywords: Civil engineering, Photographic processing, Quantitative methods, Automation, Deep Learning

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to the Ministry of Higher Education Malaysia for their generous financial support, which made this research possible. My heartfelt thanks also go to Universiti Tunku Abdul Rahman (UTAR) for providing excellent facilities and a conducive research environment.

I am deeply thankful to my supervisors, Ir. Ts. Dr. Lau See Hung, Dr. Chong Siaw Yah, and Prof. Dr. Yasuo Tanaka, for their unwavering guidance, patience, and encouragement throughout this journey. Your expertise and mentorship were invaluable to my academic and personal growth.

To my family, your unconditional love and support have been my greatest source of strength. To my friends, thank you for your encouragement and for standing by me through the highs and lows of this journey.

Finally, I appreciate everyone who contributed in any way to the completion of this dissertation. Your support has made a meaningful difference.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS / SYMBOLS

AI	Artificial Intelligence
ANN	Artificial neural network
ASPP	Atrous Spatial Pyramid Pooling
BN	Batch normalization
BSCS	British Soil Classification System
c	Number of output channels
CIAS	Crack Image Analysis System
CIF	Crack intensity factor
CNN	Convolutional neural network
CT	Computed tomography
DIC	Digital image correlation
DL	Deep Learning
DLv3+	DeepLabv3+
E	Error rates
FCN	Fully convolutional network
FL	Focal Loss
FN	False negative
FP	False positive
FPN	Feature pyramid network
FPS	Frames per second
g_t	Gradient vector at time step t
h	Height of feature maps
HSV	Hue, saturation, value
IoU	Intersection over Union

k	Number of channels
k'	Number of output channels
KNN	K-nearest neighbours
L_{avg}	Average crack length
L_{sum}	Total crack length
LL	Liquid limit
m_t	Update for first moment estimate at time step t
$\hat{m}_t$	Bias corrected first moment estimate at time step t
ML	Machine Learning
MP	Megapixel
n	Module repetitions
N_{int}	Number of intersections
N_n	Number of nodes
N_{seg}	Number of segments
p_t	Predicted probability
PI	Plasticity index
PIV	Particle image velocimetry
PL	Plastic limit
R_{SC}	Surface crack ratio
ReLU	Rectified linear unit
RGB	Red, green, and blue
RH	Relative humidity
ROIs	Region of interest
RPN	Region proposal network
s	Stride

SEM	Scanning electron microscopy
SPP	Spatial Pyramid Pooling
SVM	Support vector machine
t	Expansion factor
T	Threshold value
TN	True negative
TP	True positive
UAV	Unmanned air vehicle
v_t	Update for second moment estimate at time step t
$\hat{v}_t$	Bias corrected second moment estimate at time step t
w	Width of feature maps
w_i	weights coefficient
w_{avg}	Average crack width
w_{max}	Maximum crack width
W-D	Wetting and drying
$\overline{x}$	Mean value
α_t	Balancing weight
°C	Celsius degree
γ	Focusing parameter
θ_t	Parameter at time step t
β_1	Exponential decay rate for m_t
β_2	Exponential decay rate for v_t
λ	Regularization factor
σ	Standard deviation

CHAPTER 1

INTRODUCTION

1.1 General Introduction

Expansive soils are generally characterised by the presence of clay materials, which exhibit substantial volume changes in response to variations in moisture content (Jones and Jefferson, 2012). When these soils lose moisture, significant shrinkage occurs, resulting in the development of desiccation cracks that form as the soil dries (Nahlawi and Kodikara, 2006). The formation of extensive desiccation crack networks is attributed to tensile stresses generated during evaporation exceeding the tensile strength of the soil, particularly when volumetric shrinkage during drying is restricted (Kodikara and Costa, 2013). This phenomenon poses a significant challenge in various engineering domains, especially in geotechnical and geoenvironmental engineering.

In the last few decades, desiccation cracking has been widely studied to investigate the underlying mechanism, influencing factors, and the effect of this phenomenon (Corte and Higashi, 1964; Albrecht and Benson, 2001; Costa, Kodikara and Shannon, 2013; Tang et al., 2021). It has been established that desiccation cracking significantly affects the hydraulic and mechanical properties of soils, such as increased soil permeability and water infiltration capacity, and lower tensile strength after cracking (Morris, Graham and Williams, 1992; Rayhani, Yanful and Fakher, 2008; Cheng et al., 2021). These

combined effects essentially lead to deteriorated performance of expansive soil applications, including failures in slope, dam, foundation, clay liner, and agricultural production (Tang et al., 2021).

With escalating climate change bringing more extreme weather conditions, the destructive effects of desiccation cracking can be significantly amplified, making desiccation cracking analysis increasingly important (Zeng et al., 2019b). Early attempts in quantifying soil desiccation cracks involved manual measurements, which were intrusive and laborious methods that yielded inaccurate results (Dasog and Shashidhara, 1993). The turn of the century with its advancements in computer technology transformed the analysis method to non-intrusive image analysis techniques. In the image analysis approach, image segmentation through image processing algorithms was extensively applied to produce segmented masks with crack networks (Vogel, Hoffmann and Roth, 2005; Tang et al., 2008; Liu et al., 2013; Shit, Bhunia and Maiti, 2015; Lu et al., 2016; Singh, Rout and Tiwari, 2018; Yang et al., 2022). However, the use of image processing techniques often requires heavy customisation of the processing pipeline on a case-by-case basis to produce a satisfactory segmented crack mask. This limited the practical usage of the method in large-scale projects with larger image datasets. Therefore, the integration of artificial intelligence for soil crack quantification, particularly deep learning algorithms, is an emerging trend that offers a more accurate and automated analysing system (Han et al., 2022; Xu et al., 2022b; Pham, Ha and Kim, 2023).

In this study, laboratory experiments mimicking natural weathering conditions were conducted on expansive soils. The desiccation crack networks formed were collected using image acquisition methods. The collected images were processed and prepared as datasets for training deep learning-based segmentation models. Additionally, a series of image processing techniques was designed and employed on segmented masks to quantify crack geometrical parameters. The detection performance of various segmentation methods was evaluated through evaluation metrics scoring systems and error rate computations.

1.2 Importance of the Study

The study of desiccation cracking in expansive soils is crucial due to its widespread impact on various engineering and environmental applications. Desiccation cracks can lead to severe damage in infrastructure, including buildings, roads, and embankments, due to differential settlement in soils and structural fatigue (Cheng et al., 2020; Tang et al., 2021). The cyclic swelling and shrinking of expansive soils worsen the impacts, posing severe safety risks and maintenance costs to these infrastructures. Moreover, desiccation cracks alter the soil's hydraulic properties, which may lead to soil erosion and water infiltration problems that can turn into serious environmental issues such as contamination by nutrients and leachates leaching (Bronswijk, 1991; Albrecht and Benson, 2001; Cheng et al., 2020).

As expansive soils are omnipresence, the development and implementation of advanced monitoring technologies on desiccation cracking are essential. Image-based techniques and deep learning algorithms offer a promising solution for accurate and efficient quantification of desiccation cracks from small to large scales scenarios. By providing detailed quantification of crack patterns, the information obtained can enhance understanding of cracking mechanisms and aid in design and maintenance practices in the engineering field.

1.3 Problem Statement

Due to the loss of water through evaporation, expansive soils contract and form desiccation cracks on their surfaces as they dry. In the field of engineering, desiccation cracking significantly alters soil properties and compromises structural integrity. The presence of cracks in near-surface soils degrades their hydraulic and mechanical properties, posing safety issues to many geo-related structures (Morris, Graham and Williams, 1992; Rayhani, Yanful and Fakher, 2008; Wang et al., 2018; Cheng et al., 2020).

Numerous experimental techniques have been employed to observe and describe the soil desiccation cracking process. Various parameters have been defined to measure and quantify the geometrical characteristics of crack morphology. Combining these approaches, cracking process could be analysed both qualitatively and quantitatively (Tang et al., 2021). However, there is still no standard technique or procedure for soil desiccation testing, making it

difficult to conduct a comparative study or validate certain findings. In addition, majority of the quantitative studies of concern employed traditional image processing techniques for crack image analysis, which required extensive human intervention and effort to quantify the crack network with limited accuracy (Vogel, Hoffmann and Roth, 2005; Tang et al., 2008; Lu et al., 2016; Singh, Rout and Tiwari, 2018; Al-Jeznawi, Sanchez and Al-Taie, 2021). The traditional crack analysing system prompted the need for a more innovative and automated process for soil desiccation cracking study.

1.4 Research Questions

In response to the challenges identified in existing methods for desiccation cracking quantification and analysis, this study aims to address the following questions:

1. How can a standardised experimental setup be effectively designed to capture expansive soil crack images for desiccation crack analysis?
2. Can deep learning based segmentation methods accurately and efficiently recognise desiccation cracks using soil crack images?
3. Can the geometrical characteristics of crack morphology be effectively quantified through image processing techniques?

1.5 Aim and Objectives

The aim of this research is to explore the feasibility and effectiveness of image-based techniques in quantifying desiccation cracking behaviour of

expansive soil. The research objectives designated to be fulfilled are outlined below:

i. To design the experimental setup for desiccation crack image acquisition.
ii. To evaluate crack imaging analysis based on deep learning algorithm.
iii. To quantify desiccation cracking through image processing techniques.

1.6 Scope and Limitation of the Study

This study focused on the application of image-based techniques and crack imaging analysis, particularly with deep learning algorithms, to quantify desiccation cracking in expansive soils. The scope includes conducting laboratory experiments to capture crack patterns, developing, and validating deep learning models for crack detection and segmentation, applying image processing techniques for crack geometrical characteristics quantification, and assessing the accuracy and reliability of these methods. However, the study was limited to kaolinite clay and controlled laboratory conditions. Simulation of field conditions and applications, variations in soil types, and real-world environmental factors imitation were beyond the scope of this research due to practical constraints. Additionally, the size of the training data and its quality obtained through the experiments performed constrained and influenced the effectiveness of the deep learning models in this study.

1.7 Contribution of the Study

This study contributed to the field of engineering, especially geotechnical and geoenvironmental, by introducing innovative image-based techniques based on deep learning algorithms for the quantification of desiccation cracking in expansive soils. The research provided a comprehensive framework for capturing and analysing crack patterns, offering a more accurate and efficient method compared to traditional approaches, which are typically labour-intensive and highly sensitive to noise and lighting variations.

Unlike conventional methods, deep learning algorithms learn to recognise cracks directly from raw image data, enabling robust detection of complex crack geometries with minimal human intervention. This breakthrough addresses the limitations of traditional crack analysis approaches by providing automated and adaptive solution for soil crack quantification and recognition.

The findings can enhance understanding of desiccation cracking mechanisms by introducing the use of automated crack analysis process that potentially leads to the development of standardised techniques for crack quantification with high accuracy. Besides, the use of deep learning models for segmenting crack networks with image processing techniques which quantify important crack parameters such as surface crack ratio and crack width, can be utilised to develop on-site crack recognition system. This kind of system can aid in slope analysis to investigate the effect of desiccation cracks on slope collapse during the rainy season. The integration of deep learning algorithms with image processing techniques also represented a significant advancement in the

application of artificial intelligence in geo-related research as it provides a highly automated segmentation pipeline.

1.8 Outline of the Report

The study of deep learning-based image segmentation for expansive soil desiccation crack recognition and quantification is presented in 5 chapters. Chapter 1 provided a general introduction to the study, outlined the importance of the research, the existing problem on the issue, and detailed the aim and objectives of the research. The scope and limitations, including the contribution of the study are also highlighted in this chapter. Chapter 2 reviewed existing literature on expansive soils, desiccation cracking, traditional methods of crack analysis, and recent advancements in image-based techniques and deep learning algorithms for soil crack segmentation and detection. Chapter 3 presented the methodology used for the study, detailing the experimental setup, data collection methods, segmentation methods used (traditional and deep learning methods), implementation of the segmentation methods, and image processing techniques for crack analysis. Chapter 4 presented the findings from the experiments and analysis. The effectiveness of the traditional and deep learning models on crack segmentation was discussed under various scenarios. Lastly, Chapter 5 summarises the key findings with highlights on their implications to the engineering field. Recommendations for future research and practical applications are also provided in the last chapter.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Expansive soils present major challenges in geotechnical engineering due to desiccation cracks that compromise structural integrity. This literature review presents an overview of the phenomenon of expansive soil desiccation cracking, including general mechanisms of crack generation, the connection between expansive soils and crack generation, and the consequences of this phenomenon. Additionally, key factors influencing cracking behaviour are examined. The focus of this review is on the quantification studies of soil desiccation cracks. Traditional image processing methods for detecting soil cracks are investigated, followed by a review of the integration of machine learning and deep learning techniques in automated soil desiccation crack recognition and segmentation.

2.2 Expansive Soil Desiccation Cracking

Expansive soil desiccation cracking is a critical phenomenon with significant implications for various geotechnical applications, particularly concerning the stability and integrity of structures constructed on such soil. A comprehensive understanding of the mechanisms underlying desiccation crack formation, the relationship between expansive soils and cracking behaviour, the

factors influencing crack development, and the consequences of such a phenomenon is essential for developing effective soil management and mitigation strategies.

2.2.1 Soil Desiccation Crack Generation

Desiccation cracking in soil is generated through a series of interconnected processes, including evaporation, volumetric shrinkage, stress accumulation, and crack formation (Tang et al., 2021). As water evaporates from the soil, negative pore pressure develops, creating an attractive force between soil particles that increases the tensile stresses within the drying soil matrix. These tensile stresses drive the volumetric shrinkage of soil (Kodikara and Costa, 2013). Desiccation cracks begin to form when the accumulated tensile stresses exceed the soil's tensile strength, as illustrated in Figure 2.1 (Tang et al., 2021). The process begins with the initiation of micro-cracks, which subsequently propagate and coalesce into larger, visible cracks as drying progresses (Wang et al., 2018).

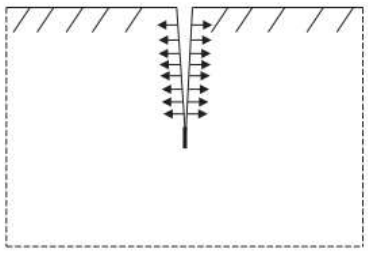


Figure 2.1: Tensile failure leading to crack initiation (Tang et al., 2021).

2.2.2 Link Between Expansive Soil and Crack Generation

Expansive soils exhibit substantial volume changes as a result of changes in moisture levels. These soils swell when water is absorbed and shrink during drying, with this behaviour primarily driven by the presence of clayey materials. Clay materials consist of fine-grained particles with a high plasticity index and a low shrinkage limit. These characteristics cause them to undergo excessive volume changes with varying water content (Jones and Jefferson, 2012). The volumetric shrinkage of clay materials contributes to the development of desiccation cracks in expansive soils, as the associated volume changes induce tensile stresses that lead to crack initiation and propagation (Morris, Graham and Williams, 1992; Tang et al., 2008; Tang et al., 2011b).

2.2.3 Consequences of Desiccation Cracking

The consequences of desiccation cracking in expansive soils are manifold, impacting the engineering properties of soils, environmental stability, and the integrity of structures built on or using them. Desiccation cracks significantly alter the hydraulic properties of soils, increasing their hydraulic conductivity by several magnitudes, which can lead to issues such as soil erosion and differential settlement (Albrecht and Benson, 2001; Rayhani, Yanful and Fakher, 2008). Besides, Cheng et al. (2021) demonstrated that desiccation cracks enhance water infiltration into the soil, resulting in rapid moisture loss and increased shrinkage rates during the drying process. This exacerbates the formation and propagation of cracks, further compromising soil

structure and integrity. Additionally, cracking reduces the mechanical strength and stability of soils, rendering them more vulnerable to deformation and failure under loads (Morris, Graham and Williams, 1992).

From an environmental perspective, desiccation cracks facilitate water infiltration and significantly increase hydraulic conductivity, allowing irrigation water containing nutrients to flow out of the root zone rapidly (Bronswijk, 1991). This process not only deprives crops of essential nutrients but also increases the risk of eutrophication in downstream water bodies, such as lakes, due to nutrient leaching. Additionally, desiccation cracking in clay structures, such as landfill barriers, creates preferential flow paths for hazardous leachates to escape, thereby contaminating the surrounding environment (Cheng et al., 2020).

In the engineering field, desiccation cracks pose severe risks to buildings and infrastructures. The cyclic swelling and shrinking of expansive soils can lead to differential settlement and structural fatigue, resulting in the formation of cracks in foundations, structural elements, and pavements (Cheng et al., 2021). In addition, the increased water retention capacity on slopes due to the presence of surface cracks promotes soil erosion and reduces soil cohesion. This combination of detrimental effects compromises slope stability and can ultimately result in slope failures (Wang et al., 2018).

2.3 Factors Influencing Cracking Behaviour

The desiccation cracking behaviour of expansive soils is closely related to and dependent on the surrounding conditions to which the soil is exposed. Factors such as temperature, drying and wetting cycles, soil composition and structure, and boundary conditions significantly influence the initiation and propagation of cracks.

2.3.1 Temperature

Temperature plays a vital role in the desiccation cracking behaviour of expansive soils. Tang et al. (2010) investigated the temperature-dependent behaviour of expansive soils by subjecting clay specimens with an initial water content of 170 % to drying temperatures of 22 °C, 60 °C, and 105 °C. The main findings, summarised in Table 2.1, demonstrated that higher temperatures accelerate the evaporation rate of soil moisture, leading to more extensive cracking, as indicated by the increased surface crack ratio. Additionally, the study revealed that cracking begins at higher soil water content when the soil is exposed to elevated temperatures. Several studies further indicate that the tensile and structural strength of clayey soils decreases with rising temperature (Gu et al., 2014; Salimi et al., 2021). This reduction in strength at higher temperatures reduces the soil's resistance to tensile stresses, resulting in earlier crack initiation at higher moisture levels.

Table 2.1: Desiccation tests under different drying temperatures (Tang et al., 2010).

Specimen	Temperature (°C)	Initial evaporation rate (g/min)	Surface crack ratio (%)	Cracking water content (%)
S1	22	0.021	13.9	35.7
S2	22	0.022	13.8	40.1
S3	22	0.022	14.2	38.2
S4	60	0.252	17.3	79.1
S5	60	0.255	16.9	74.8
S6	60	0.250	16.9	72.3
S7	105	0.507	23.1	96.1
S8	105	0.501	21.5	87.1
S9	105	0.499	22.3	89.5

2.3.2 Drying and Wetting Cycles

The cyclical nature of drying and wetting significantly influences crack formation and behaviour in expansive soils. During drying periods, soils experience volumetric shrinkage, which leads to crack development. When wetting occurs, these cracks typically close; however, repeated drying and wetting cycles often exacerbate crack development. Several studies have demonstrated that this weathering process intensifies the formation of cracks

over time (Yesiller et al., 2000; Tang et al., 2011a; Al-Jeznawi, Sanchez and Al-Taie, 2020).

Yesiller et al. (2000) measured the crack extent using the crack intensity factor (CIF), which represents the proportion of the crack area relative to the total soil surface area. Their findings revealed that CIF increased with the number of drying and wetting cycles, indicating that the crack severity intensified as the soil underwent repeated cycles. Similarly, in their investigation of clay soil cracking under drying and wetting cycles, Tang et al. (2011a) observed that the surface crack ratio (another term for CIF) also increased with the number of cycles, reaching equilibrium after approximately three cycles, as shown in Figure 2.2. Additionally, they observed that crack pattern stabilised after a few drying and wetting cycles, with most cracks forming in the same locations as in the previous cycles.

Al-Jeznawi, Sanchez and Al-Taie (2020) reported similar stabilisation of crack patterns, suggesting that desiccation cracks formed during drying cycles created weak planes in the soil structure. Although wetting cycles may close these cracks, the weakened bonds at these sites facilitated easier crack initiation, increased horizontal shrinkage, and promoted the formation of more microcracks and a higher surface crack ratio.

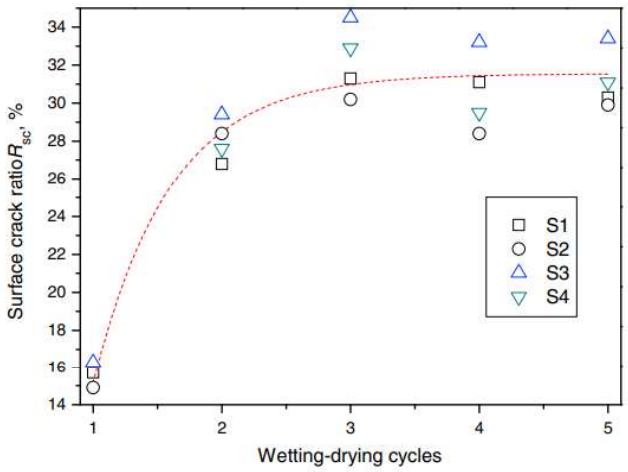


Figure 2.2: Surface crack ratio versus number of drying and wetting cycles (Tang et al., 2011a).

2.3.3 Soil Composition and Structure

The composition and structure of the soil significantly affect its susceptibility to desiccation cracking. The shrink-swell properties of expansive soils are strongly influenced by their clay content and mineral composition. Different types of clay minerals exhibit different water-retention capacities and expansiveness due to their distinct mineral structures. For instance, montmorillonite, which is held together by weaker van der Waals forces, contrasts with kaolinite, where the layers are bonded by stronger hydrogen bonds. As a result, montmorillonite has a higher water retention capacity and greater shrink-swell potential, owing to the ease with which its bonds can separate (Budhu, 2010).

Several studies have demonstrated the effect of soil composition on the desiccation behaviour of expansive soils (Albrecht and Benson, 2001; Tang et

al., 2008; Al-Jeznawi, Sanchez and Al-Taie, 2020; Shepidchenko et al., 2020). Albrecht and Benson (2001) reported higher volumetric shrinkage in specimens with a higher smectite content, which can be attributed to smectite's primary composition of montmorillonite. Tang et al. (2008) made a similar observation in their study of shrinkage cracks in clayey soils, noting increased CIF and crack width with a higher plasticity index. In a comparative study, Al-Jeznawi, Sanchez and Al-Taie (2020) examined kaolinite and bentonite mixtures and found that mixtures with a higher plasticity index required more wetting and drying cycles to reach an equilibrium state during desiccation tests compared to a lower plasticity pure kaolinite mixture. This finding suggests that soils with higher swelling clay content exhibit greater instability. Similarly, Shepidchenko et al. (2020) recently investigated the factors controlling desiccation cracks using montmorillonite, illite, and kaolinite. They observed increased crack width and earlier crack initiation in samples with higher montmorillonite content, which possesses the highest shrink-swell potential among the three minerals.

Aside from soil mineral composition, the initial structural configuration of the soil particles can significantly influence cracking behaviours during the desiccation process. Several studies have demonstrated that soil samples prepared using slurry and compaction methods exhibit distinct desiccation cracking behaviours (Corte and Higashi, 1964; Albrecht and Benson, 2001; Cheng et al., 2020). Early experimental investigations by Corte and Higashi (1964) suggested that compaction densifies the soil structure, reducing the evaporation rate, resulting in slower crack initiation and propagation in

compacted soil samples. Similarly, Albrecht and Benson (2001) found that increased compaction effort decreased volumetric shrinkage strain, indicating that compaction minimises soil deformation during the desiccation process. In a more recent study, Cheng et al. (2020) used soils compacted at different water contents, including on the dry side and wet side of optimum water content, respectively. They reported that dry-compacted soils exhibited simultaneous crack initiation across the soil surface, with more uniformly distributed cracks. In contrast, wet-compacted soils initially developed a primary crack, followed by secondary cracks much later in the drying process. These findings underscore the significant role of soil's initial configuration in determining its desiccation cracking behaviour.

2.3.4 Boundary Conditions

Boundary conditions, such as geometry, bottom interface, and the presence of vegetation, significantly influence crack formation. Several studies have investigated the impact of sample geometry by varying sample area and thickness (Prat, Ledesma and Lakshmikantha, 2006; Nahlawi and Kodikara, 2006; Tang et al., 2008). Prat, Ledesma and Lakshmikantha (2006) examined the effect of sample size and found that mean crack width and soil clod area decreased as the sample area increased. For the effect of soil thickness, Nahlawi and Kodikara (2006) observed that mean crack width and mean soil clod area increased with greater sample thickness. They also reported that thicker soil layers exhibited slower desiccation rates due to the increased distance moisture

travelled to the surface. Similarly, Tang et al. (2008) reported analogous observations in their experiments with clay specimens of various thicknesses.

Soils constantly interact with various materials in nature, and these interactions can significantly influence their desiccation cracking behaviour. Laboratory experiments have provided evidence of this phenomenon by introducing different interface frictions through boundary manipulations (El Hajjar et al., 2019; Zeng et al., 2019a; Al-Jeznawi, Sanchez and Al-Taie, 2021). El Hajjar et al. (2019) found that kaolinite clays developed cracks under all interface conditions, ranging from smooth to rough. Zeng et al. (2019a) observed that cracks propagated faster with increasing interface roughness when soil thickness remained constant, and the amplitude of this effect decreased with greater soil thickness. They also reported that increased interface roughness or reduced soil thickness led to more severe crack formation. Similarly, Al-Jeznawi, Sanchez and Al-Taie (2021) recorded higher CIF values with greater bottom interface friction.

In addition to bottom constraints, expansive field soils are often covered by vegetation, which can act as a boundary constraint. Gao, Zeng and Shi (2021) explored the effects of vegetation on the stability of red clay slopes with desiccation cracking during rainfall infiltration. Their findings indicated that vegetation, particularly with deep tap root system, significantly improved slope stability by restricting desiccation crack development. On the other hand, Cheng et al. (2023) studied how soil desiccation cracking behaviour is affected by varying vegetation densities. They found that vegetated soils exhibited a

smaller final surface crack ratio and narrower average crack width compared to non-vegetated soils. Additionally, higher planting densities were associated with more simultaneous crack initiation, but with finer and thinner crack lines. This suggests that desiccation cracking can be mitigated to a certain extent by optimising vegetation density.

2.4 Quantification Studies of Soil Desiccation Crack

Expansive soil desiccation crack analysis has become an increasingly important area in numerous engineering fields due to its implications on infrastructures and geotechnical structural safety. This has driven the development of quantitative approaches for analysing desiccation cracks. Such analyses use fractal geometry to characterise cracks, providing insights into the process of fracture formation and its extent. Early manual soil characterisation methods included wire probes, cement slurry, and dye (El Abedine and Robinson, 1971; Li et al., 2019; Zhao and Koseki, 2020). Although these approaches were cost-effective and easily accessible, they often disrupted the crack pattern and subsequently reduced the accuracy of quantitative measurements (Liu et al., 2013).

The evolution of computer science and image digitisation in the twentieth century introduced the field of computer vision, encompassing subfields such as image analysis and image processing. As this field matured, many soil crack quantitative studies shifted to non-intrusive image analysis and image processing techniques (Liu et al., 2013; Shit, Bhunia and Maiti, 2015;

Singh, Rout and Tiwari, 2017; El Hajjar et al., 2019; Al-Jeznawi, Sanchez and Al-Taie, 2020). Image analysis involves extracting information from digital images through a series of processes, including image acquisition, image processing, and feature extraction, and then ensembles the information to perform object recognition (Gonzalez and Woods, 2008). In soil desiccation crack recognition and quantification, image analysis separates the crack as the subject of interest, commonly referred to as the foreground, from the non-crack area, known as the background. This separation facilitated accurate quantitative characterisation of cracks.

Over the years, researchers have defined various geometrical parameters to describe the soil crack morphology. The concept of using CIF as an indicator of the extent of soil surface cracking was first introduced by Miller, Mi and Yesiller (1998). CIF is defined as the time-variable ratios of the total crack area at a given time to the soil's total surface area at initial time (time zero). To enhance clarity, Tang et al. (2010) proposed an alternative name for the parameter CIF as surface crack ratio (R_{SC}). Both CIF and R_{SC} have since been widely used in soil crack studies as they help researchers in determining the correlation between crack intensity and the various factors influencing soil cracking behaviour (Tang et al., 2010; Tang et al., 2008; Shit, Bhunia and Maiti, 2015; Lu et al., 2016; Bamgbopa, 2016; Singh, Rout and Tiwari, 2018; Zeng et al., 2019b; Al-Jeznawi, Sanchez and Al-Taie, 2021; Cheng et al., 2021; Yang et al., 2022).

In addition to crack extension, other geometry parameters, such as average crack width, crack length, and clod properties, have been defined to further describe soil desiccation cracking behaviour. Liu et al. (2013) developed a software tool called “CIAS”, which stands for Crack Image Analysis System. The software automatically characterises soil cracks through a pipeline of operations, including image segmentation, crack recognition, and geometric parameter measurement. For the software to generate accurate results, input soil crack images must be captured under well-controlled lighting and angles. CIAS was specifically designed for soil desiccation crack detection. However, many related studies still rely heavily on general-purpose image processing programs, such as ImageJ, making the quantification analysis tedious, inconsistent, and lacking automation (Tang et al., 2021). To address these limitations, machine learning methods are increasingly applied in image segmentation tasks to ensure better performance and robustness in crack detection (Xu et al., 2022b; Han et al., 2022; Pham, Ha and Kim, 2023).

2.4.1 Image Acquisition

To perform image analysis, digitised images must first be obtained. Several acquisition methods and tools have been employed in research to capture representations of desiccation cracking networks. Commonly used techniques include digital image correlation (DIC), particle image velocimetry (PIV), laser instruments, X-ray computed tomography (CT), scanning electron microscopy (SEM), and cameras (Tang et al., 2021).

Digital Image Correlation (DIC) is a useful technique for measuring surface displacement and strain fields. It generates contour maps of displacement and strain by capturing and calculating pixel deformations between an initial image (before displacement) and a subsequent image (after displacement) (Wei et al., 2016). Using DIC, Wei et al. (2016) demonstrated that expansive soil cracking typically follows three distinct modes, including opening, sliding, and tearing. The opening mode is the most prevalent, often occurring alone or in combination with sliding or tearing modes. They also observed that cracks generally propagate perpendicularly to the direction of tension, although the presence of shear strains can alter the propagation direction (Wei et al., 2016). In a comparative study on desiccation cracking behaviour of kaolinite and montmorillonite clay, El Hajjar et al. (2019) used DIC to monitor the local strain evolution. Their findings revealed that montmorillonite exhibited a deformation threshold twice that of kaolinite before cracking occurred, highlighting the material's greater resistance to strain-induced cracking.

Li et al. (2019) introduced tracer particles into fluids and adopted the particle image velocimetry (PIV) method to track and measure continuous velocity fields within the fluid. Costa, Kodikara and Shannon (2013) utilised PIV analysis to study factors affecting clay desiccation cracking, observing the evolution of tensile strain during shrinkage-induced cracking. Additionally, PIV has been coupled with DIC for tensile strain analysis on clayey soil, demonstrating that tensile stress distribution can be used to predict crack location and propagation in real time (Li et al., 2019). Figure 2.3 shows an

example of the soil tensile failure process in three stages (rising stress, failure developing, post-failure stage) and their relative DIC and PIV images at specific points on the tensile deformation curve (Figure 2.4) to highlight their correlation. These visualisations help to bridge the relationship between deformation mechanism and cracking behaviour. While DIC and PIV are highly effective for deformation analysis, offering precise displacement and strain data, they are less practical for applications focused on efficiently quantifying crack geometry. Both methods require high-resolution imaging equipment and consistent imaging conditions to ensure accuracy. Moreover, DIC and PIV rely on specialised software for results interpretation, making them computationally intensive and time-consuming, limiting their feasibility for real-time monitoring applications.

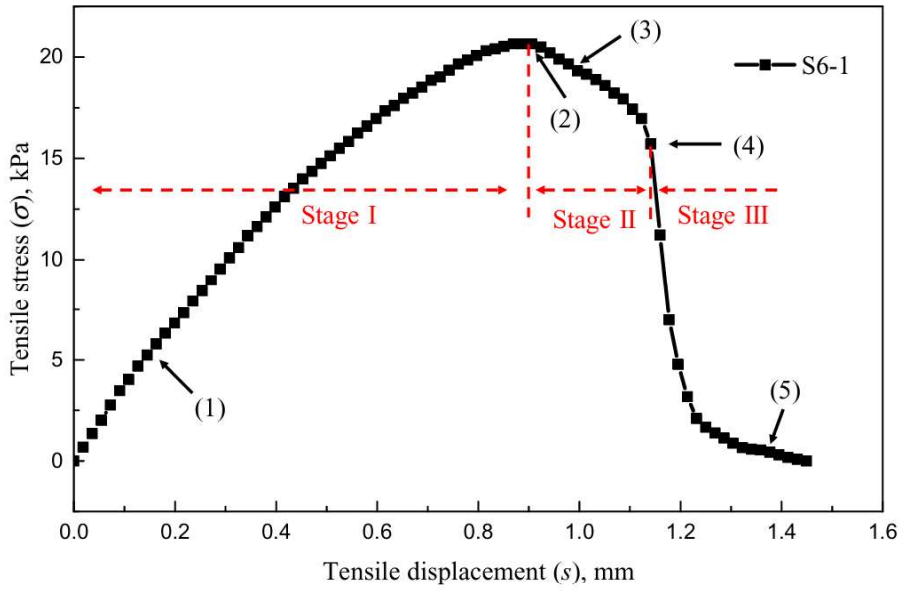


Figure 2.3: Soil tensile deformation curve (Li et al., 2019).

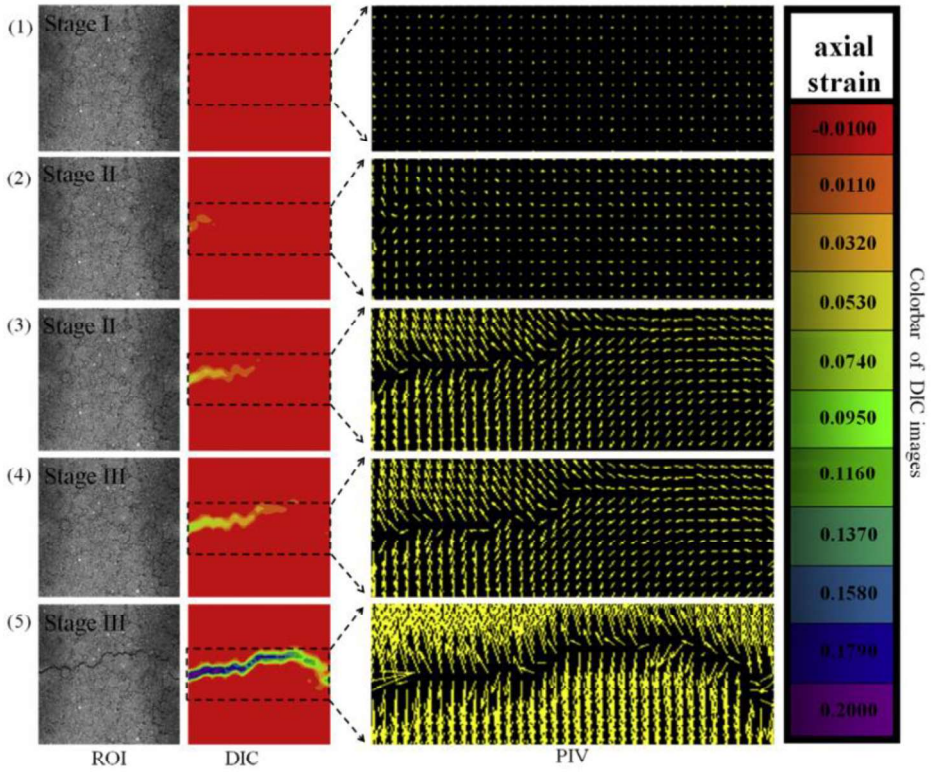


Figure 2.4: DIC and PIV images for different tensile stages at ROI (Li et al., 2019).

To examine subsurface cracks in soil, researchers have employed advanced methods like laser technology and X-ray computed tomography (CT). These non-destructive imaging techniques enable three-dimensional measurements of crack networks beneath the soil surface. X-ray CT, in particular, has proven effective in visualising the geometric and volumetric properties of buried crack networks (Tang et al., 2021). Although these techniques offer valuable insights for 3D quantitative analysis of crack networks, they require expensive equipment and involve extensive post-data analysis efforts. In addition, their reliance on specialised instruments limits their application primarily to laboratory-scale studies. Consequently, there has been

a growing trend toward adopting image-based acquisition methods that involve capturing images of soil crack networks and analysing them for fast and cost-effective way (Mohan and Poobal, 2018; Munawar et al., 2021).

Among the characterisation methods, a setup involving a digital camera, a supporting frame, and a light source has emerged as a cost-effective and efficient solution for image acquisition setup (Wang et al., 2018). The speed of post-processing is proportional to the resolution of the captured images, with higher resolution requiring greater computational resources. Traditionally, the captured images are analysed using image processing techniques, which involve steps like pre-processing and segmentation to extract crack-related information. However, in recent years, the rapid advancements in artificial intelligence and computer vision have led to the increased use of machine learning and deep learning methods for the recognition and quantification of soil desiccation cracks. A detailed discussion of traditional image processing techniques and novel deep learning approaches will follow in the subsequent sections.

2.5 Crack Detection with Traditional Image Processing Technique

Conventionally, image processing plays two major roles in the image analysis pipeline: image enhancement and image segmentation, which precede feature extraction and characterisation (Hadjiiski, Samala and Chan, 2021). Traditional image processing techniques have successfully fulfilled these roles by performing low-level operations such as noise reduction and contrast enhancement to optimise image quality, as well as mid-level processes such as

segmentation, which separates regions of interest from the background (Gonzalez and Woods, 2008). Inspired by its successful application in medical imaging analysis, researchers have applied these techniques to crack detection in soils.

Early examples of soil crack detection using traditional image processing techniques relied heavily on manual intervention during characterisation (Vogel, Hoffmann and Roth, 2005; Peng et al., 2006). Vogel, Hoffmann and Roth (2005), for instance, utilised a variable thresholding method to segment grey-scale images. Grey-scale images are monochromatic, consisting solely of light intensity values ranging from black to white, as shown in Figure 2.5 (Gonzalez and Woods, 2008). In their study, Vogel, Hoffmann and Roth (2005) set the segmentation threshold value at each location in the image to 65% of the mean intensity value within a surrounding bounding box of 20 × 20 pixels. Thresholding works by dividing the intensity values of an image into two distinct groups using a predefined threshold value, effectively separating the image into a binary format. In a binary image, pixel intensity is represented as 0 for the background and 1 for the object of interest (Gonzalez and Woods, 2008). However, this approach faced significant limitations. The selection of the threshold value was subjective and lacked generalisation, resulting in inconsistent outcomes and limited applicability.

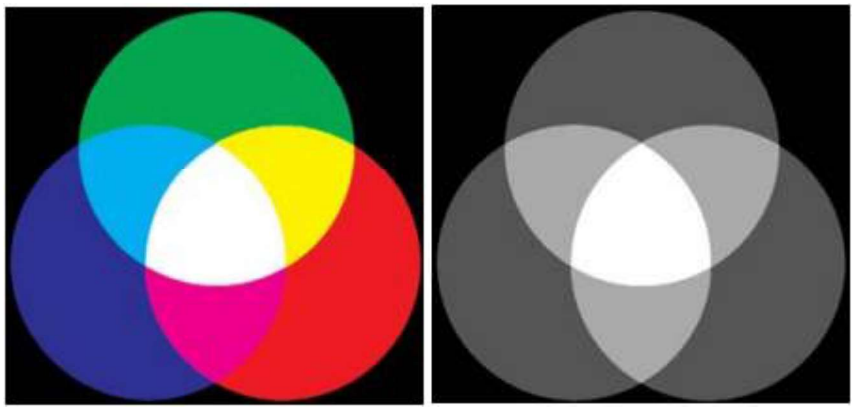


Figure 2.5: RGB image (left) and intensity image (right) (Gonzalez and Woods, 2008).

Peng et al. (2006) used Adobe Photoshop's "Magic Wand" function to segment crack images and measured the surface crack area using an open-source software called Scion Imaging. The "Magic Wand" tool selects image areas based on colour similarity, but this approach often generates noise (pseudo-cracks) that require manual elimination. As a result, their characterisation method was highly impractical for larger datasets due to the substantial human effort involved.

CIAS is a soil quantification system using image processing techniques by Tang et al. (2008). The system produced binary segmented crack images with calculated geometrical parameters, including the number of intersections, crack segments, crack width, crack length, clod area, and surface crack ratio. In their paper, Tang et al. (2008) conducted tests on low-plasticity clay to explore the relationships between drying temperature, soil thickness, wet-dry cycles, and their effects on desiccation cracking behaviour. Building on this foundation work, the same research group later carried out more detailed investigations into

individual factors influencing desiccation cracking (Tang et al., 2010; Tang et al., 2011a; Tang et al., 2011b).

The image processing techniques used in CIAS and its workflow were later detailed by Liu et al. (2013). As mentioned previously, CIAS consists of three primary procedures: image segmentation, crack identification, and parameter measurement. An iterative clustering-based method was employed to automatically determine the global threshold value, *T*, for each image during the segmentation step. The process begins by selecting an initial threshold value, *T*, which is set to the average grey level of the image. The image is then segmented into two groups of pixels: white and black. Pixels with grey intensity values exceeding *T* are classified as white, whereas those with values below *T* are labelled as black. Next, the mean grey intensity values of each group are calculated, and a new threshold *T* is computed as the average of these two mean values. This process is repeated iteratively, updating *T* in each iteration until the difference between consecutive *T* values becomes negligible. The final *T* represents the optimal threshold value for the given image. This method, as described by Gonzalez and Woods (2008), is a standard global thresholding approach. However, they noted that this technique performs effectively only when the histogram of the image exhibits a distinct valley between the crest corresponding to the object and the background. After segmentation, the resulting image typically contains noise (black spots) and discontinuous crack lines with white dots as shown in Figure 2.6 (c). To address these issues, Liu et al. (2013) proposed the use of closing operation to eliminate the white dots and using the seed-filling algorithm for black spot removal. Closing is a

morphological operation that smoothens contours by removing small holes, filling gaps, and connecting small breaks (Gonzalez and Woods, 2008). Figure 2.6 shows the original colour crack image, after segmentation, and spot removal. Seed filling algorithm identifies individual regions of interest by starting with a seed pixel and iteratively adding neighbouring pixels that meet predefined connectivity or intensity criteria (Yu, He and Xi, 2010). By setting a threshold value for the area of black spots and white dots, the algorithm identifies and removes regions that satisfy this condition, resulting in a cleaner image, as demonstrated in Figure 2.7 (Tang et al., 2008).

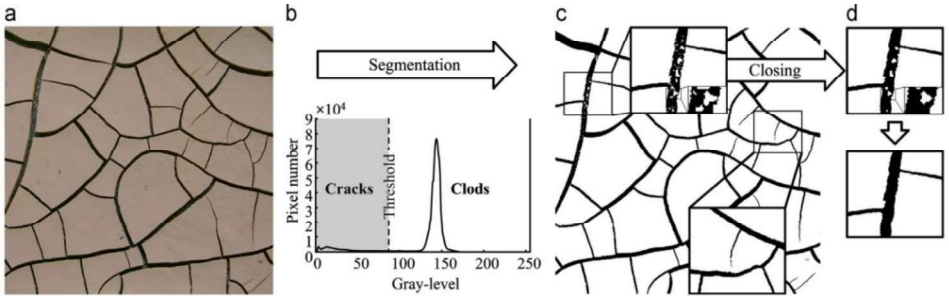


Figure 2.6: Process from original image to segmentation with thresholding and closing operation for spots removal (Liu et al., 2013).

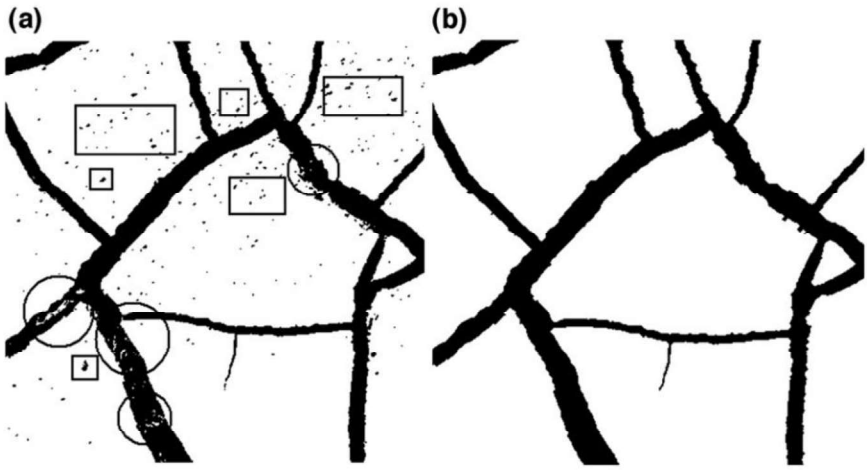


Figure 2.7: Spot and noise removal. (a) Original segmented image, and (b) after seed-filling spot removal method (Tang et al., 2008).

Recent studies have explored the applicability of CIAS quantification system. Cheng et al. (2020) investigated the effect of compacted soil microstructure on its cracking behaviour under varying optimum water content. CIAS was employed to measure crack density (defined as the number of crack segments per unit area) and surface crack ratio. These parameters revealed that changes in compaction water content during sample preparation significantly alter the soil's microstructural state. Furthermore, the trajectories of surface crack ratio and crack density increments varied depending on those states. The same group of researchers conducted another study on the effect of drying and wetting cycles on soil infiltration capacity (Cheng et al., 2021). They utilised CIAS to calculate the surface crack ratio and determined that a threshold of 4% marked the point at which the surface crack ratio begins to drastically affect the infiltration capacity. Similarly, in an investigation of the dependency of soil cracking behaviour on relative humidity (RH), Zeng et al. (2022) reported that crack width increased with higher RH, while total crack length increased with

lower RH. These findings underscore the utility of CIAS in quantitatively analysing soil cracking behaviour across various environmental and preparatory conditions.

Despite its advantages, the CIAS quantification system is not without limitations. The effectiveness of its global thresholding method heavily relies on the quality of input images. However, the major downside of this system is that the input images are highly quality-restricted. For instance, uneven illumination and noise can result in poorly defined crest-and-valley structures in the image histogram, rendering the thresholding process ineffective. As a result, achieving accurate results often requires well-controlled imaging conditions, which can limit its practical applicability in certain scenarios.

The aforementioned CIAS was mostly automatic in the sense that when an image was fed to the system, an output was generated through the processing pipeline with minimal user intervention. Aside from CIAS, most of the existing quantitative studies relied on open-source image processing software such as ImageJ for image segmentation, while others used MATLAB for subsequent analysis. Image segmentation typically consisted of converting RGB images into grey-scaled images, then performing segmentation with various thresholding methods to obtain the binary result (Shit, Bhunia and Maiti, 2015; Lu et al., 2016; Bamgbopa, 2016; Singh, Rout and Tiwari, 2018; El Hajjar et al., 2019; Al-Jeznawi, Sanchez and Al-Taie, 2021; Yang et al., 2022). In some instances, image enhancement or post-processing steps were included to achieve better segmentation of soil cracks, thereby enabling finer geometrical

parameter measurements (Singh, Rout and Tiwari, 2018; Al-Jeznawi, Sanchez and Al-Taie, 2021; Yang et al., 2022).

In a demonstration of the use of image processing techniques in image segmentation, Lu et al. (2016) determined the optimal global threshold value for each image by using Otsu's method, implemented in MATLAB via the "graythresh" function. Otsu's method identifies the ideal threshold by maximising the variance between two classes of pixels that represent foreground and background pixels, respectively (Gonzalez and Woods, 2008). Among the studies that utilise image processing techniques, only Lu et al. (2016) specified their binarisation method. This showed the lack of a standardised thresholding technique, and the choice of binarisation procedure largely depends on the attributes of the specific image under investigation and the image processing application or software available.

On the other hand, several studies demonstrated the use of image processing techniques in pre- or post-processing of binary images to improve segmentation outcomes. Al-Jeznawi, Sanchez and Al-Taie (2021) applied rolling ball background subtraction on grey-scaled images to mitigate the effect of background intensity variations during binarisation. In this case, the rolling-ball algorithm estimates the background intensity of an image and subtracts it from the grey-scaled image, resulting in a more uniform background. After binarisation, the researchers used the "Outline" option in ImageJ to obtain the crack boundaries by selecting the crack areas as regions of interest. Figure 2.8 shows the image analysis procedures performed with ImageJ software.

Similarly, Yang et al. (2022) adopted homomorphic and Gaussian filter (smoothing filter) as pre-processing steps prior to binarisation. Homomorphic filtering adjusts the illumination and reflectance components in an image to enhance the visibility of lower intensities and sharpened edges (Gonzalez and Woods, 2008). The Gaussian filter was used for denoising which further improved the image quality for subsequent processing.

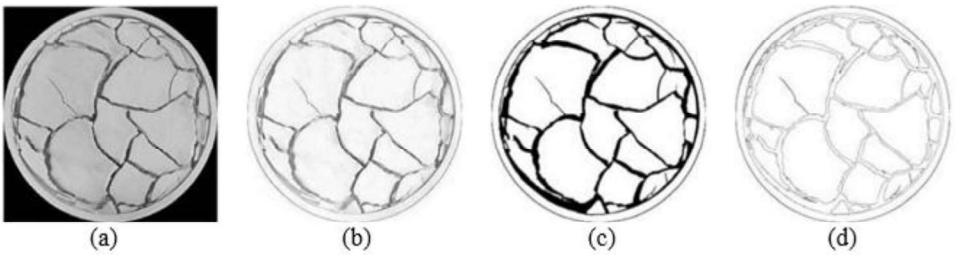


Figure 2.8: Main analysis processes. (a) Grey-scaled image, (b) after background subtraction, (c) binarisation, and (d) outlining boundaries (Al-Jeznawi, Sanchez and Al-Taie, 2021).

In a study investigating the effects of soil initial properties on its desiccation behaviour, Singh, Rout and Tiwari (2018) performed grey scaling and segmentation with ImageJ software and subsequent crack quantification in MATLAB. To enhance the quality of the binary image, they applied a closing operator and median filtering to remove noises that are similar to those described in CIAS. Median filtering is particularly effective in reducing speckle noise by substituting the intensity of a pixel with the median intensity value of its neighbouring pixels (Gonzalez and Woods, 2008). Quantification analysis, such as the determination of CIF and crack width, was then performed on MATLAB. Prior to quantification analysis, freehand selection operator was

used to manually select the cracks within the container to determine the crack area.

Overall, these studies illustrated the role of image processing techniques in soil crack detection. However, many of the procedures lack sufficient specificity, with limited details provided on the methods used. Moreover, these studies rely heavily on human intervention, requiring individual processing of images which introduces potential inconsistencies and subjectivity in the analysis.

2.6 Machine Learning

Machine Learning (ML), as a subfield of Artificial Intelligence (AI), employs computational algorithms to solve statistical problems by making predictions based on input data (Issam and Murphy, 2015). Unlike traditional programming, a machine learning algorithm does not require explicit instructions to perform a task; instead, it configures its architecture automatically through iterative analysis of patterns and inferences from data. This process, often referred to as training, enables the algorithm to make predictions or decisions with minimal human intervention. The widely recognised definition of machine learning, attributed to Arthur Samuel, described it as “a field of study that gives computers the ability to learn without being explicitly programmed” (Issam and Murphy, 2015; Xin et al., 2018).

Typically, machine learning is categorised by its approaches during learning, and the three broad categories are supervised, unsupervised, and reinforcement learning (Janiesch, Zschech and Heinrich, 2021). Supervised learning uses a labelled dataset for algorithm training, where the dataset consists of input data paired with their corresponding output labels (Issam and Murphy, 2015). Whereas unsupervised learning uses unlabelled data and allows the algorithm to independently identify patterns and correlations within the inputs (Issam and Murphy, 2015). Unlike these approaches, reinforcement learning trains the algorithm by enabling it to interact with a given environment and receive feedback through rewards or penalties. The aim is to optimise cumulative rewards to achieve a predefined objective. Reinforcement learning is commonly applied in areas such as gaming, robotic control, and autonomous systems, where sequential decision-making is essential (Janiesch, Zschech and Heinrich, 2021).

Due to its remarkable ability to address classification and regression problems, machine learning has been increasingly utilised in numerous fields, including image and speech recognition, financial scenario analysis, natural language processing, and medical diagnosis (Issam and Murphy, 2015; Janiesch, Zschech and Heinrich, 2021; Xin et al., 2018). Some widely recognised machine learning algorithms, including linear regression, K-nearest neighbours (KNN), and support vector machine (SVM), were created to analyse data and make reasonable predictions (Han et al., 2022). In a surface crack monitoring study, Zhang et al. (2021) proposed the use of support vector machine (SVM) in conjunction with F-score feature selection to detect and extract surface cracks

from unmanned air vehicle (UAV) images. SVM is a powerful machine learning algorithm and classifier that looks for an optimal hyperplane that divides observations of different classes from one another by leveraging patterns of information called features (Pisner and Schnyer, 2020). F-score calculates the significance of different features and helps in the selection of the most relevant features that enable more efficient predictions and classifications between classes (Tao et al., 2013). Zhang et al. (2021) proved their model's ability to detect the presence of cracks in the UAV images with an overall accuracy rate of 89.5%. However, the use of machine learning was restricted to the crack's existence detection, and the method of segmentation or extraction of cracks was not described. The advantages of implementing machine learning in this case are limited as it only classifies images as crack or non-crack images.

Traditional machine learning methods face challenges when processing raw natural data due to their inability to identify patterns without a feature extractor. Designing a machine learning system capable of recognising patterns requires significant domain expertise to create a feature extractor that pre-processes raw data into formats suitable for the learning subsystem to analyse and identify the patterns (LeCun, Bengio and Hinton, 2015; Janiesch, Zschech and Heinrich, 2021). To reduce the dependence on domain expertise, deep learning was developed, inspired by the functioning of human neurons. Deep learning employs multiple artificial neural networks to perform computations on vast amounts of data, enabling automated pattern recognition without extensive manual feature engineering (LeCun, Bengio and Hinton, 2015).

2.7 Deep Learning

Subsequent research into AI technologies led to the emergence of deep learning (DL), a specialised subset of machine learning. DL is a form of representation learning that uses artificial neural networks (ANNs) to address high-dimensional problems through multiple levels of abstraction. The term "deep" refers to the multi-layered structure of the artificial neural network (Xin et al., 2018). Mimicking human biological neural networks, ANNs consist of artificial neurons organised into layers, where each neuron in one layer is interconnected with every neuron in the next layer through weighted connections, forming a deep neural network. In representation learning, the system automatically identifies relevant features from input data for detection or classification tasks without requiring a manually designed feature extractor. To achieve multiple levels of abstraction, non-linear modules (layers) are combined, with each module responsible for transforming the current data representation into a more abstract level (LeCun, Bengio and Hinton, 2015).

Consider an image with pixel values as an example, the first layer of a deep learning system may learn to recognise edges at specific positions within the image. This information is then passed to the second layer, which might detect patterns formed by these edges, even if their positions are slightly altered. As the process continues, deeper layers learn to combine these patterns into larger, more complex structures, representing parts of recognisable objects. Eventually, the system can detect entire objects based on these hierarchical combinations (LeCun, Bengio and Hinton, 2015). In deep learning, the system

identifies patterns from extracted features using a general-purpose learning technique. By automating the identification of hierarchical representations, deep learning enables the machine to understand complex patterns in the input without requiring manually designed feature layers. Among DL, Convolutional neural networks (CNNs) have emerged as one of the most successful algorithms, excelling in the recognition and segmentation of objects and regions within images (LeCun, Bengio and Hinton, 2015; Han et al., 2022).

A comparative analysis of traditional machine learning models, such as SVM, and deep learning models, such as CNN, demonstrated that deep learning models have better recognition accuracy in image classification, particularly when applied to large-scale datasets (Wang, Fan and Wang, 2021). CNNs are a type of feedforward neural network capable of extracting features directly from raw data through their convolution structures. These structures consist of multiple arrays of convolutional layers, making CNNs well-suited for processing both 2D data, such as images, and 3D data, like videos (LeCun, Bengio and Hinton, 2015).

In general, CNN architecture comprises three types of layers: a convolutional layer, followed by a pooling layer, and finally a fully connected layer (O'Shea and Nash, 2015). The convolutional layer performs feature extraction from input data using learnable filters and a non-linear activation function is often used after this. Rectified linear unit (ReLU) is an example of this activation which introduces non-linearity into the output. Subsequently, pooling layers are employed to down-sample the spatial dimensions of the input,

preventing overfitting and computational complexity. After multiple stacks of convolutional layers, activation units, and pooling operations, the fully connected layer is positioned at the last layer in a CNN architecture. This layer maps the output of the preceding layer to an output label (Munawar et al., 2021; O'Shea and Nash, 2015). An illustration of the combination of these layers to form a CNN architecture is shown in Figure 2.9.

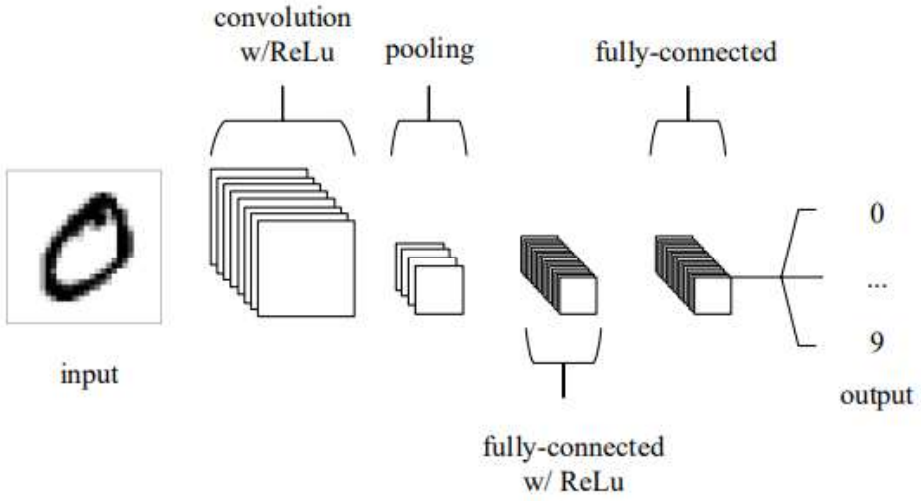


Figure 2.9: A simple five-layer CNN classification model (O'Shea and Nash, 2015).

Another example of a convolutional network is illustrated in Figure 2.10, where an image of a Samoyed dog is fed into the network. Each horizontal rectangle in the row represents a feature map (LeCun, Bengio and Hinton, 2015). Feature maps are the outputs of convolutional layers, with each map corresponding to a specific learned feature. These maps are generated by applying convolutional operations with filters (or kernels). The output indicates that the class Samoyed has the highest class score based on the detected features,

allowing the network to classify or predict the image as belonging to the Samoyed class.

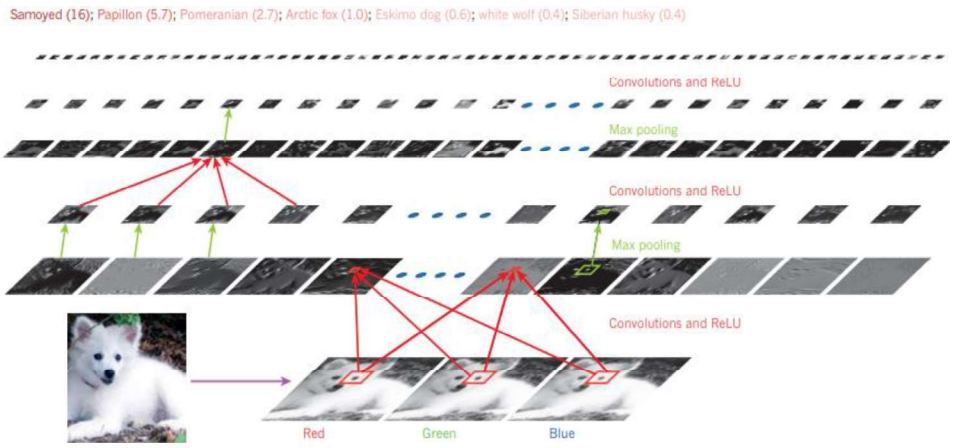


Figure 2.10: Bottom up following how a CNN classify an image (LeCun, Bengio and Hinton, 2015).

2.7.1 Soil Crack Detection Using Deep Learning

Recent studies on soil crack detection using deep learning and CNNs have demonstrated significant success, achieving high efficiency and accuracy (Xu et al., 2022b; Han et al., 2022; Andrushia et al., 2022; Pham, Ha and Kim, 2023; Xu et al., 2022a; Xu et al., 2024). The application of deep learning on soil crack recognition was first introduced by Xu et al. (2022a), who utilised a U-Net convolutional neural network for detecting soil desiccation cracks in laboratory-acquired images. The proposed U-Net architecture is shown in Figure 2.11.

As per its name, U-Net architecture features a symmetrical U-shaped design, comprising an encoder and a decoder. The encoder, located on the left, captures features while progressively reducing the spatial dimension of the input image. The decoder, on the right, restores the image's spatial dimensions and generates the final segmentation mask, which in this case is a binary mask. The "Copy" operation shown in Figure 2.11 represents skip connections that transfer feature maps directly from each encoder layer to its corresponding decoder layer. These skip connections enhance object localisation by combining the encoder's low-level spatial features with the decoder's high-level information.

The performance of the model was evaluated through comparison of precision, recall, and Dice scores between U-Net model and traditional thresholding methods. When assessed against the ground truth, the U-Net model achieved precision, recall, and Dice scores of 94.38%, 74.43%, 81.13%, compared to 81.74%, 66.56%, 68.48% for the thresholding method. These results conclusively demonstrated that deep learning using U-Net significantly improves the segmentation process in soil crack detection.

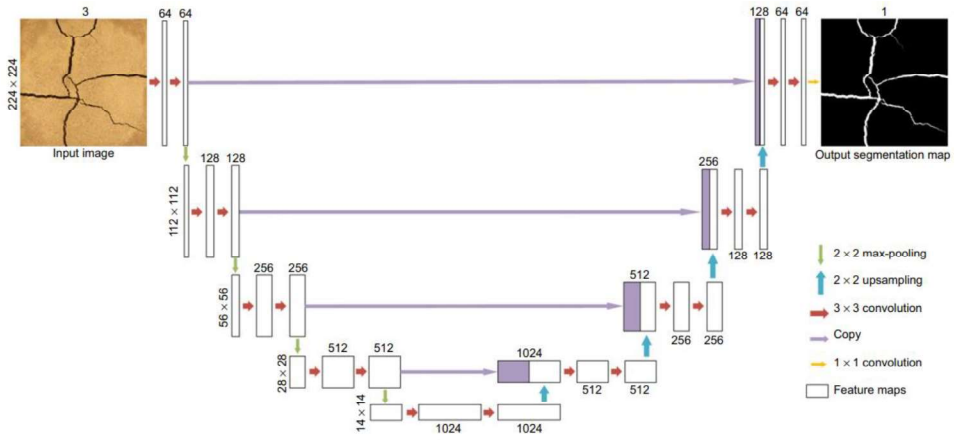


Figure 2.11: U-Net architecture for semantic segmentation of soil desiccation crack image (Xu et al., 2022a).

Despite the promising performance of the U-Net model, the same group of researchers identified that the model lacks robustness in handling images captured under uneven illumination conditions (Xu et al., 2022b). To address this limitation, Xu et al. (2022b) developed a new soil crack recognition system called Attention Res-UNet for semantic segmentation of crack networks. The structure of the Attention Res-UNet incorporates deep Res-UNet architecture enhanced with attention gates, as shown in Figure 2.12. Deep Res-UNet shares a structural similarity with conventional U-Net but differs in its integration of residual connections. The term 'Res' in Res-UNet refers to the residual connection, which allows input information to bypass intermediate layers and connect directly to their outputs. This design facilitates the effective training of deeper networks by mitigating the vanishing gradient problem. To counter the effects of uneven illumination, the researchers introduced an attention gate in the model. The attention gate selectively emphasises critical information in the image while filtering out irrelevant responses, thereby

enhancing segmentation accuracy. This improvement makes Attention Res-UNet a more robust and effective tool for crack detection under varying illumination conditions.

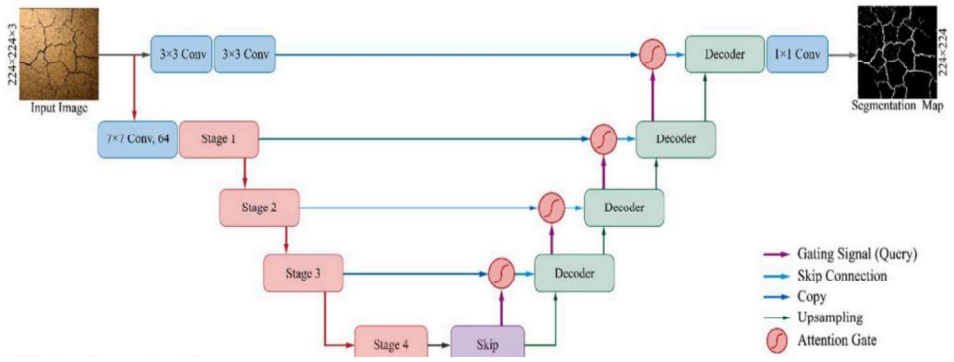


Figure 2.12: Structure of Attention Res-UNet (Xu et al., 2022b).

The performance of the proposed Attention Res-UNet model was evaluated against seven other deep learning models designed for semantic segmentation tasks (Xu et al., 2022b). The segmentation results of all models are illustrated in Figure 2.13. It is evident that Res-UNet and Attention Res-UNet produced the most satisfying outcomes based on visual assessment. The evaluation metrics used to quantitatively evaluate the performance of different models were precision, recall, and the Jaccard index (or Intersection over Union, IoU). The Jaccard evaluates the similarity between the predicted positive and true positive by dividing their intersection by their union. The higher values indicate better segmentation performance, as they reflect a higher proportion of correctly identified instances located accurately. From Table 2.2, the Attention Res-UNet achieved the highest scores in precision and Jaccard index, outperforming other deep learning models and traditional segmentation

methods. However, while the inclusion of attention gates improved segmentation accuracy as illustrated in Figure 2.13 (i) and (k), Table 2.2 suggests that the improvement was minimal. The Res-UNet alone was effective in mitigating the effects of uneven illumination, indicating that the primary contribution of the attention gate was relatively subtle in these scenarios.

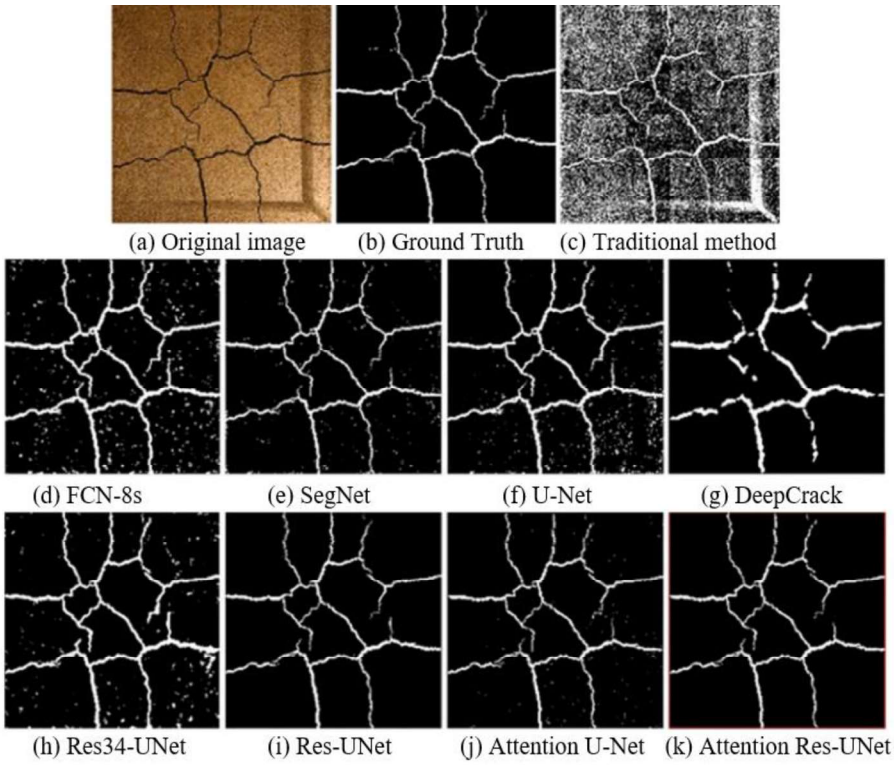


Figure 2.13: Final segmentation map of different models (Xu et al., 2022b).

Table 2.2: Performance of different models under different evaluation metrics (Xu et al., 2022b).

Models	Precision (%)	Recall (%)	Jaccard (%)
Traditional Method	13.27	47.1	11.51
FCN-8s	56.56	93.44	53.99
SegNet	71.88	79.16	60.32
U-Net	64.19	88.23	58.46
DeepCrack	28.18	83.73	26.61
Res34-UNet	59.53	90.32	55.45
Res-UNet	80.4	77.84	65.6
Attention U-Net	79.16	74	61.73
Attention Res-UNet	81.28	78.15	66.68

In the same year, Han et al. (2022) published a crack detection and localisation system based on Mask R-CNN. By leveraging transfer learning, a dataset consisting of 1200 annotated crack images was used to train the model. Mask R-CNN is a cutting-edge deep learning algorithm designed for both image segmentation and object detection. Unlike U-Net which generates a semantic segmentation map where all pixels belonging to the same class are assigned a class label, Mask R-CNN performs instance segmentation that assigns unique labels to individual instances within the same class, in addition to the general class label. The flow structure of the Mask R-CNN model is illustrated in Figure 2.14. In Stage 1, feature maps were extracted from the input images using the ResNet-50 CNN backbone. A region proposal network (RPN) then scanned the feature maps to identify rectangular boxes of Regions of Interest (ROIs) that

potentially contain cracks. The ROIs were then aligned with the feature maps in the align layer to eliminate spatial misalignments. In Stage 2, the aligned feature maps were passed through a fully convolutional network (FCN) that simultaneously generated a segmentation mask and a class label with its bounding box which indicates the crack location.

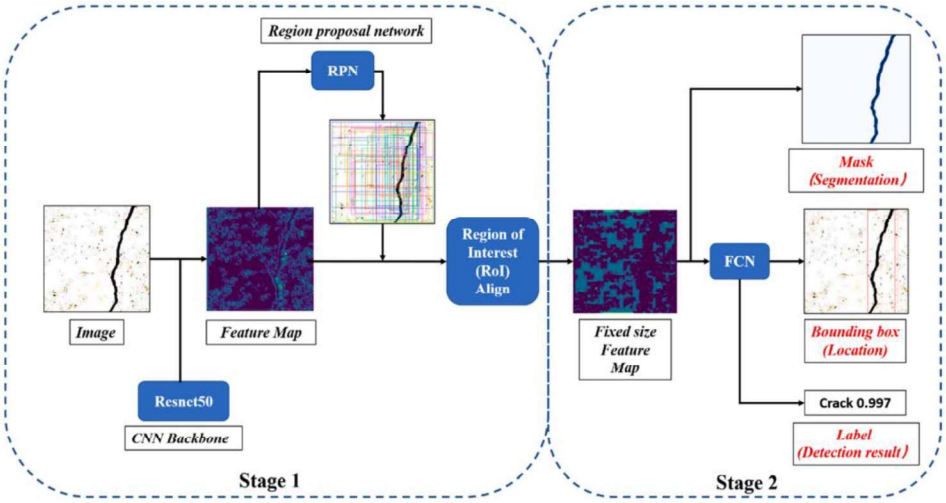


Figure 2.14: Architecture of Mask R-CNN model (Han et al., 2022).

The performance of the proposed Mask R-CNN algorithm was evaluated by using common DL evaluation metrics which are precision, recall, and F1 score, achieving 73.3%, 82.8%, and 77.7%, respectively. Additionally, the authors compared the performance of Mask R-CNN with U-Net under varying degrees of soil background complexity. An example of detection and segmentation results for both models is illustrated in Figure 2.15. Under all tested conditions, Mask R-CNN outperformed U-Net in crack detection, localisation and segmentation. The authors tested the performance of Mask R-CNN on solid soil backgrounds (least complex), impure backgrounds

(containing scratches with slightly uneven soil surface), and complex crack scenarios (high-density crack networks). It was reported that although the algorithm provided satisfactory detection in all cases, its accuracy declined as background complexity increased. Despite this, Mask R-CNN consistently demonstrated superior performance compared to U-Net in handling complex crack images. This study highlights the potential of Mask R-CNN as an effective instance segmentation deep learning algorithm for automated soil desiccation crack recognition. Its ability to detect and segment individual crack instances in complex scenarios makes it a robust candidate for soil cracking analysis.

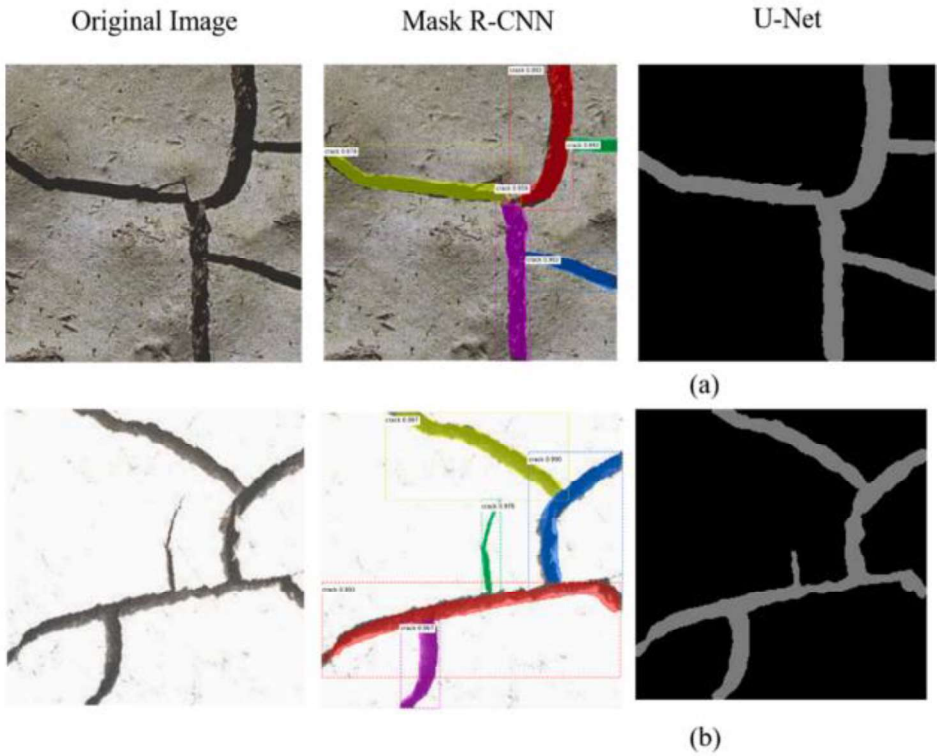


Figure 2.15: Crack detection results from different models (Han et al., 2022).

In another study, Andrushia et al. (2022) introduced a deep-learning framework for detecting expansive soil shrinkage cracks by comparing two customised CNN networks against pre-trained VGG-19 and ResNet-50. Their dataset consisted of 5,200 field soil crack images and they employed five-fold cross-validation technique to demonstrate that their customised CNN model featuring 5 convolutional layers, coupled with 3 ReLUs and 2 fully connected layers, was able to outperform deeper pre-trained networks like VGG-19 and ResNet-50 at a precision of 0.957 and recall of 0.980. They also demonstrated that their best-performing customised CNN model was superior in terms of precision and recall when compared to traditional methods like the Canny edge detector and more advanced approaches, including SVM, DCNN, and U-Net. However, the robustness of the models in this study to variable illumination remained untested as their images were captured under relatively uniform lighting. Furthermore, by resizing the dataset to just 64 × 64 pixels, the experiments may oversimplify crack structures which might cause the evaluation metrics to not reflect the actual generalisation ability of the models in real-world scenarios.

Pham, Ha and Kim (2023) conducted a comparative study to evaluate the efficacy of ground crack detection systems between convolutional neural networks and image processing techniques. This study evaluated several popular CNN algorithms, including U-Net, DeepLabv3, LinkNet, and Feature Pyramid Network (FPN), against Otsu's segmentation method. The dataset used for CNN training is composed of images from slope experiments conducted in laboratory and field crack images from sites of high landslide occurrence. Field

images are generally complicated due to uncontrollable image acquisition and uneven surfaces when compared to laboratory-generated images. Results showed that CNN models significantly outperformed Otsu's segmentation method across all evaluation metrics: with higher precision (0.88 – 0.90), recall (0.87 – 0.92), and F1 score (0.88 – 0.90), while Otsu's method achieved 0.71, 0.70, and 0.65, respectively. The segmentation maps generated by each technique are demonstrated in Figure 2.16, which highlights the superior crack detection capabilities of CNNs, particularly in handling clear test datasets. Despite their high performance on the test dataset, deep learning models faced challenges when applied to images with complex backgrounds caused by shadows. These conditions sometimes led to reduced detection of precision. The study suggested that further research and optimisation could enhance the performance of DL models in such scenarios. On the downside, the images used in the study were mostly captured at oblique angles which distort geometric relationships, complicating subsequent crack width measurement. This necessitated field calibration for each image to obtain accurate crack dimensions, making the process impractical for large datasets.

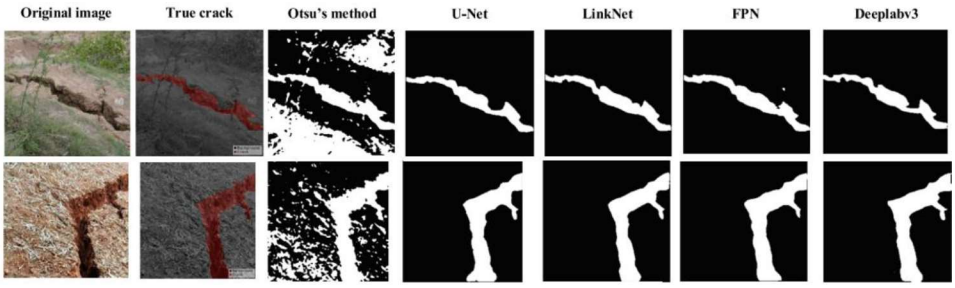


Figure 2.16: Segmentation results from Otsu's method and CNN models (Pham, Ha and Kim, 2023).

Xu et al. (2024) introduced U-CrackNet, the first unsupervised segmentation pipeline for soil-desiccation cracks that fuses simple threshold-based masks (weak knowledge) with outputs from Meta AI's Segment Anything Model (strong knowledge) to automatically generate pseudo-labels for training a lightweight U-Net (Lite-UNet). They proposed a Multi-Head Perception (MHP) module which consisted of dilated convolutions at several scales, and a Multi-Head Perception Attention (MHPA) module to blend low- and high-level features alongside the pseudo-masks to reap the benefits from both sides. Upon testing with 12 images at 1000 × 1000 pixel resolution, U-CrackNet delivered a Jaccard index 24 % higher than a traditional thresholding method and 5 % higher than Attention Res-UNet, while reducing quantification errors in surface crack ratio, total crack length, and crack segment count by roughly a factor of three. Its key strengths lie in the elimination of manual labelling and lightweight architectures for real-time deployment on resource-limited devices. Despite eliminating manual annotation, U-CrackNet's reliance on SAM for pseudo-label generation incurs substantial GPU memory and runtime overhead, which might contradict the original intention for lightweight deployment. Moreover, the model's performance on more diverse crack morphologies and lighting conditions remained untested as their dataset was captured under fairly uniform lighting.

2.8 Summary

Quantifying soil cracks is essential for understanding desiccation behaviour, as it enables the identification of influencing factors by correlating

cracking geometrical parameters. Besides understanding the cracking mechanism, insights into desiccation behaviour are crucial for several applications. In geotechnical engineering, crack quantification helps in assessing the stability of slopes, foundations, and embankments, which can be impacted by soil shrinkage behaviour. Understanding desiccation behaviour can be practical in agriculture as cracks affect water retention capacity and irrigation efficiency in soil. In addition, the study of desiccation behaviour is essential in environmental engineering, such as the use of landfill liners and clay barriers, where cracks can lead to hazardous leakage. Traditional quantification methods, such as image processing techniques, have demonstrated limitations in accuracy and efficiency, particularly when applied to complex datasets. These challenges emphasise the growing need for advanced crack detection systems. Recent advancements in deep learning methods have shown significant potential to address these limitations, offering enhanced precision and efficiency for soil crack quantification and analysis. A comparison of traditional image processing and deep learning-based approaches is presented in Table 2.3 to summarise their respective strengths and limitations in the context of soil crack detection.

Table 2.3: Critical comparison of traditional vs. deep learning approaches in soil crack detection.

Aspect	Traditional image processing	Deep learning-based approaches
Feature extraction	Rely on manual feature engineering	Automatically learns hierarchical features from data
Adaptability	Limited and required fine-tuning for different scenarios	High adaptability across varying image conditions if given sufficient training data
Accuracy	Highly sensitive to image condition (e.g. performance drops in noisy or complex backgrounds)	High segmentation accuracy despite varying conditions
Computational requirements	Generally lightweight	Required more computational resources (e.g. GPUs and datasets for effective training)
Data dependency	Limited dependency as features are manually extracted	High dependency as performance improves with larger (but diversified) datasets
Scalability	Limited scalability for field deployment due to manual feature extraction	Highly scalable once trained and can be applied to automated large-scale crack detection

CHAPTER 3

METHODOLOGY AND WORK PLAN

3.1 Introduction

This chapter presented the methods and techniques employed to investigate the feasibility of image-based techniques for quantifying desiccation cracking behaviour of clayey soil. The objectives were to conduct a soil desiccation experiment using image-based techniques, to perform image analysis with deep learning algorithms, and to quantify cracks using image processing techniques. To fulfil the aim and objectives, the methodology and workflow were designed as shown in Figure 3.1 and Figure 3.2, and presented accordingly.

The first two components of the methodology involved materials and sample preparation, and image acquisition tool setup, which describe the laboratory experiment setup designed for soil crack measurements. The subsequent section on data preparation and preprocessing described the preparation of image datasets for algorithm training and evaluation. The algorithms used for crack segmentation are outlined in two sections which are traditional segmentation methods and deep learning segmentation methods. The procedures and setups for fine-tuning the deep learning models are depicted in the model training and validation section. The last section presents the

evaluation of segmentation results and techniques for quantifying crack geometrical parameters.

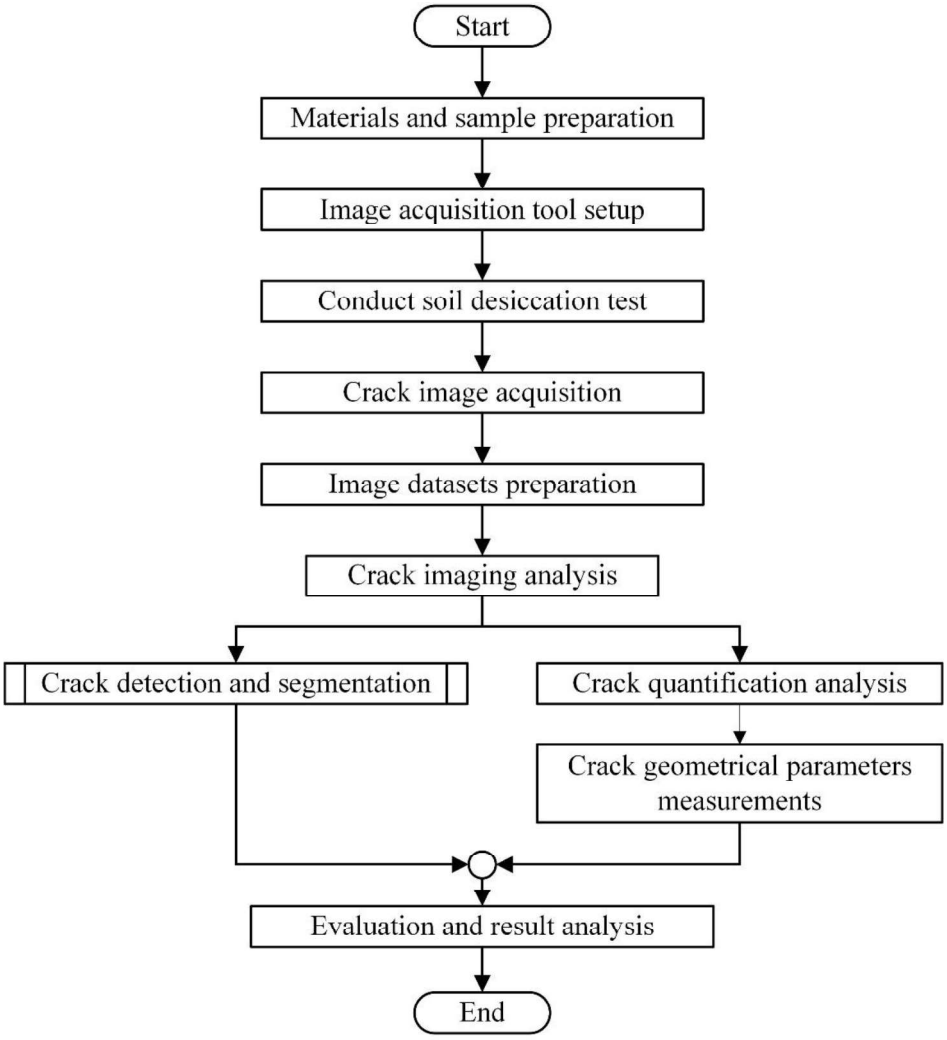


Figure 3.1: Project workflow.

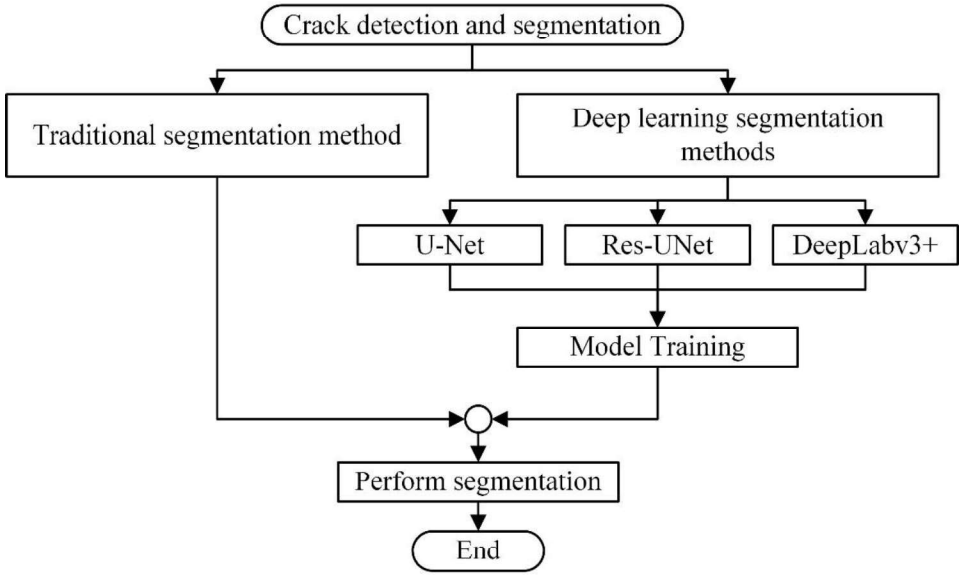


Figure 3.2: Subprocess for crack detection and segmentation.

3.2 Materials and Sample Preparation

The expansive soil used for the study was kaolin clay in powder form. Its chemical compositions and properties are presented in Table 3.1. The slurry state samples used in soil desiccation tests were prepared by mixing the kaolin clay powder with distilled water. The slurry was prepared at two water content levels which are 100% and 120%. The 100% water content corresponds to 1.5 times the liquid limit based on previous studies (Tang et al., 2011a; Tang et al., 2011b). In addition, the 120% water content was introduced to increase variability in crack morphology. To ensure a homogenous distribution of moisture, the mixture was allowed to sit for 24 hours before being transferred to the moulds (Costa, Kodikara and Shannon, 2013). The mould used for the desiccation tests was glass Petri dishes with a diameter and height of 120 mm and 20 mm. A total of 14 specimens were prepared with final settled thicknesses

of 5 mm, 10 mm, and 15 mm. This variation was intended to provide a range of crack patterns for the crack image dataset used in model training (Nahlawi and Kodikara, 2006; Tang et al., 2008). An example of a 10 mm thick specimen is shown in Figure 3.3.

Table 3.1: Kaolin clay chemical compositions and properties.

Chemical Analysis	
Aluminum (Al_2O_3)	32.0 – 38.0 %
Silica (SiO_2)	47.0 – 53.0 %
Loss on Ignition @ 1025 °C	11.0 – 14.0 %
Soil Properties	
Particle size	2.0 μm – 3.0 μm
Liquid Limit (LL)	66.0 %
Plastic Limit (PL)	36.9 %
Plasticity Index (PI)	29.1 %
BSCS Classification	MH (highly plastic silt)

Figure 3.3: 10 mm thick soil specimen.

3.3 Image Acquisition Tool Setup

To evaluate the desiccation cracking behaviour of clayey soil, an image acquisition tool was designed for soil crack measurements during desiccation tests. The tool setup is illustrated in Figure 3.4, which consists of a humidity chamber, oven bulbs (Tungsram 300 °C oven bulb 25W E14), a weighing machine (Weighing GF-10K Industrial Balance IP65 / NEMA4, 10.1 kg × 0.01 g), a humidity and temperature data logger (Multicomp Pro MP780621, ± 0.3 °C / ± 3 % RH accuracy, 0.1 °C / 0.1 % RH resolution), and a camera (SONY DSC-WX500). A humidity chamber is a thick glass chamber that restricts the fluctuation of temperature and humidity inside the chamber by limiting airflow. The schematic drawing of the chamber is shown in Figure 3.5. Soil desiccation tests were then carried out using the setup to acquire crack images throughout the desiccation process.

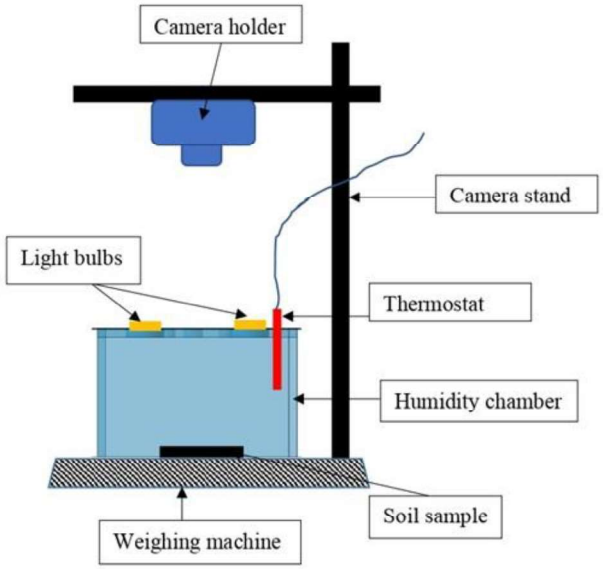


Figure 3.4: Schematic drawing for image acquisition tool setup.

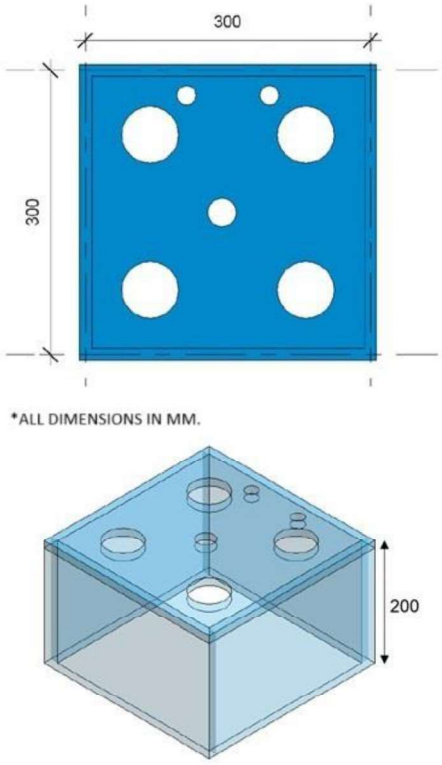


Figure 3.5: Schematic drawing for humidity chamber detailing.

3.4 Data Preparation and Preprocessing

Image dataset is a crucial component for the successful development of deep learning image segmentation model. The dataset used in this project comprised 255 crack images obtained from laboratory soil desiccation tests that simulated various field conditions on kaolinite soils using the image acquisition tool setup in Section 3.3. The original crack images were captured using a 12 MP digital camera under various photographic conditions, including variations in lighting and locations, which introduced diversity to the dataset.

In the first preparation step, the images were cropped into square pictures, excluding the mould and circumferential cracks. These square images were then resized to a uniform resolution of 960 × 960 pixels, as illustrated in Figure 3.6. From this dataset, 20 pieces at a resolution of 960 × 960 pixels were selected and designated as the test dataset for evaluating the deep learning models.

The remaining cropped images were divided into four quadrants, each with a resolution of 480 × 480 pixels, resulting in a total of 940 image patches. These patches were then manually inspected and those without cracks were discarded, as only images containing cracks are useful for the model training process. An example is demonstrated in Figure 3.7. After filtering, the final image dataset for deep learning algorithm calibration contained 800 crack images, each at a resolution of 480 × 480 pixels. Training and validation datasets

were created by randomly splitting the 800 images at an 80:20 ratio, respectively (Han et al., 2022).

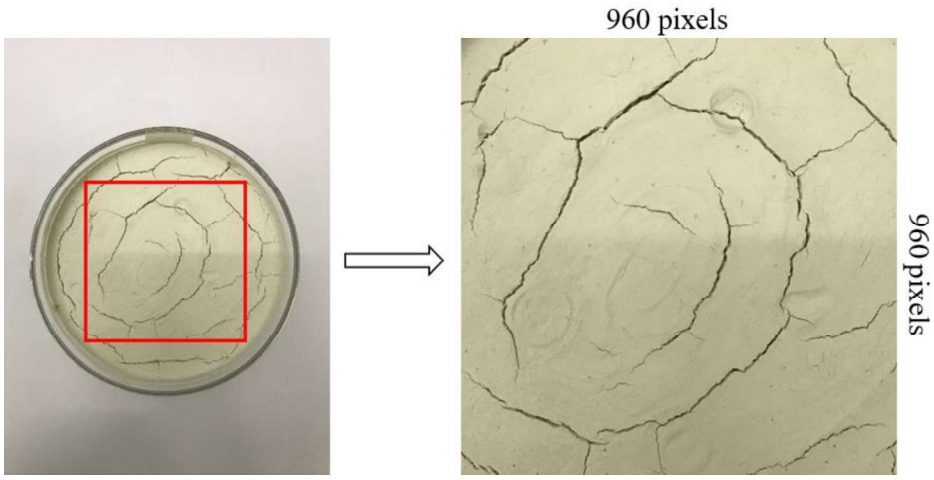


Figure 3.6: Original captured image and after cropping and resizing.

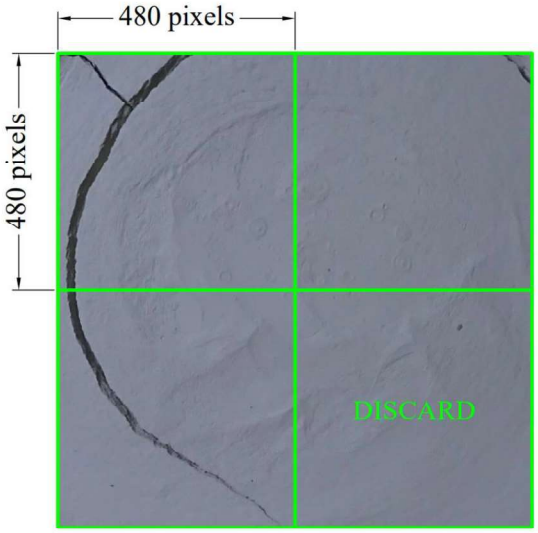


Figure 3.7: Divided image patches with non-crack patches discarded.

In this study, supervised learning was employed, where datasets are associated with their corresponding output labels. For the binary image

segmentation problem, the labelling of dataset typically involves pixel-wise classification of image pixels into foreground and background labels, represented by values of 1 (true) and 0 (false) respectively.

The images were manually annotated using Image Segmenter in MATLAB. Within the Image Segmenter, flood fill and fill holes operations were utilised to select and fill the crack regions. Binary masks were then generated, with crack pixels labelled as 1 (white) and non-crack pixels labelled as 0 (black). An example of a generated binary mask is shown in Figure 3.8.

During the subsequent preparation of datasets, the binary masks were converted to ground truth masks by mapping the relevant image pixel values to their designated class names, in this case foreground and background respectively.

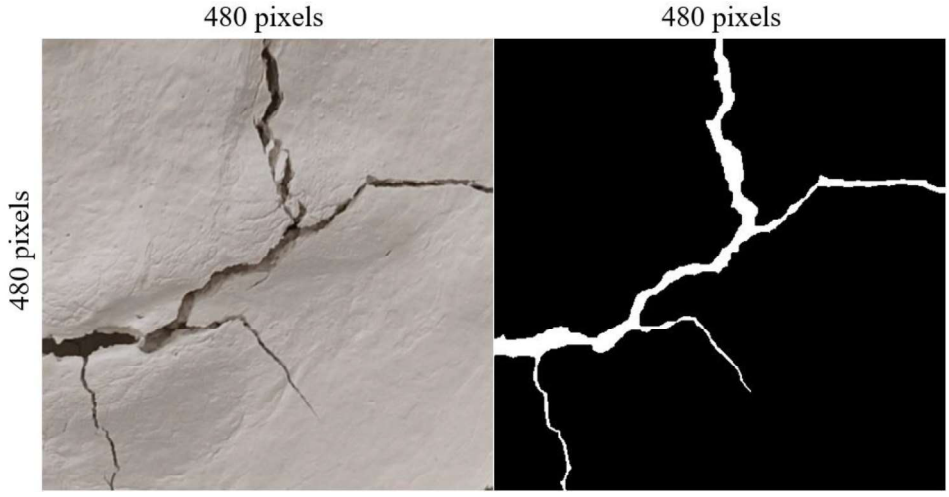


Figure 3.8: Original image patch (left) and binary mask generated by Image Segmenter (right).

The training of deep convolutional neural networks often encounters the challenge of overfitting, where the network memorises the training dataset instead of learning to capture the feature representations from the input data (Shorten and Khoshgoftaar, 2019). This issue results in inferior performance on new and unseen data. A common cause of overfitting is the lack of a sufficiently large training dataset. One effective strategy to mitigate this issue is the use of data augmentation techniques. Data augmentation enhances the dataset by applying image transformation techniques to create artificial samples from the original images, thereby promoting dataset diversification (Shorten and Khoshgoftaar, 2019).

In this study, data augmentation techniques included hue, saturation, and value (HSV) colour jittering and random 2-D transformation such as scaling, rotating, reflecting, and translating. Hue is the attribute of colour that differentiates between colours; saturation represents the purity of the colour, with higher saturation indicating less white light is mixed with the pure colour; and value is the brightness of the colour (Solomon and Breckon, 2011). The HSV colour jittering operation randomly alters the RGB image's colour by adjusting saturation, brightness, and contrast, as illustrated in Figure 3.9 (b). Additionally, 2-D transformations, such as scaling, rotating, vertical and horizontal flipping, and translating, were applied randomly to the jittered images. During each training cycle, the model was exposed to images with different augmentations applied, which helps to prevent overfitting. Examples of data augmentation are demonstrated in Figure 3.9.

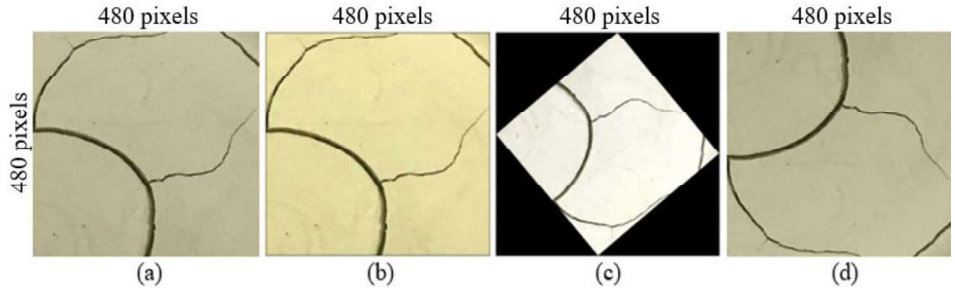


Figure 3.9: Data augmentation operations. (a) Original image; (b) HSV colour jittering; (c) Rotation; (d) Vertical flipping

3.5 Traditional Segmentation Method (Otsu's Method)

The Otsu's algorithm is a widely used segmentation method that separates foreground and background pixels in a grey-scale image by determining an optimal global threshold based on the maximisation of between-class variance (Gonzalez and Woods, 2008). The computation steps to obtain the optimal threshold value in Otsu's algorithm are as follows:

1. Compute the histogram of pixel intensities for the input image.
2. Consider all intensity values from 0 to the maximum value in the histogram as potential threshold values and iterate through them. Each threshold separates the pixels into foreground (intensity values greater than the threshold) and background (intensity values less than or equal to the threshold) pixels.
3. Compute the variances for the two classes of pixels at each threshold iteration.
4. Select the optimal threshold value as the one that maximises the between-class variance.

To obtain Otsu's segmented masks, the RGB crack images were first converted to grey-scale images using MATLAB's 'im2gray' function. Next, a built-in function 'graythresh', calculates a global threshold value using Otsu's algorithm. The binary mask was then generated by applying global thresholding on the grey-scale image with the computed threshold value using 'imbinarize' function.

3.6 Deep Learning Segmentation Models

The deep learning models used for the segmentation and recognition of expansive soil desiccation cracks in this study included U-Net, Res-UNet, and DeepLabv3+. DeepLabv3+ was implemented in four variations based on different backbone networks. The backbone networks utilised were pre-trained MobileNetV2, ResNet-18, ResNet-50, and Xception. The architectures for each model are explained thoroughly in the following sections.

3.6.1 U-Net

The convolutional network U-Net was created specifically for image segmentation tasks and was first presented by Ronneberger, Fischer and Brox (2015). Its name derives from its distinctive U-shaped architecture, which comprises an encoder, a decoder, and a bridge connecting them. The architecture of U-Net used in this study is illustrated in Figure 3.10.

The basic building block of U-Net comprises two 3 × 3 convolutional layers, with each layer followed by a ReLU activation function (Ronneberger, Fischer and Brox, 2015). This neural unit can be repeated multiple times within the encoder and decoder, depending on the desired network depth. Typically, each repetition of the basic unit contributes to one level of depth, with the encoder and decoder generally having the same levels of depth, resulting in a symmetrical structure.

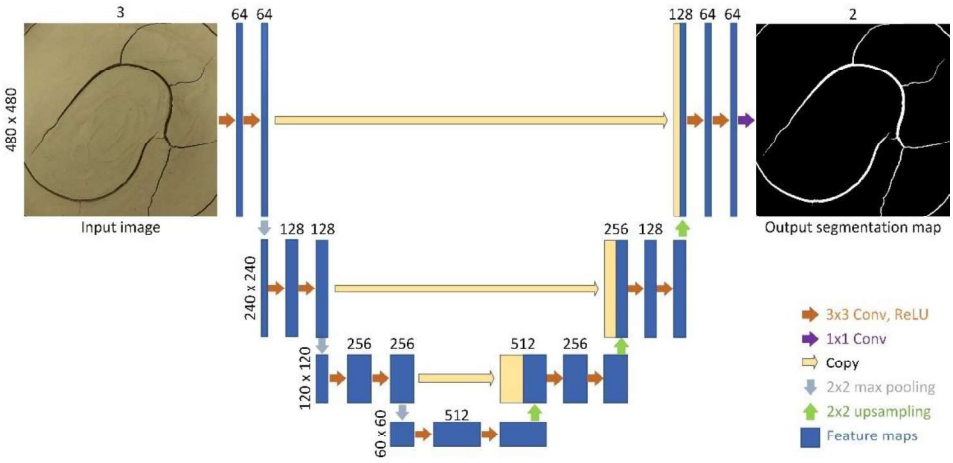


Figure 3.10: U-Net architecture. Height and width of feature map are denoted on the left of each level, with number of channels on top of each feature map box.

The encoder, on the left side, functions as a contracting path that extracts features from the input in a hierarchical manner. At the end of each encoding level, a 2 × 2 max pooling layer with a stride of 2 is used to reduce the spatial dimensions of the feature maps by half (downsampling). Concurrently, the depth of the feature maps is doubled, enabling the network to capture richer

representations and encode more information. Following the encoding process, a bridge unit is used to connect the encoder to the decoder. The bridge retains the encoded information and transfers it to the decoder, which is crucial for maintaining information flow across the network.

Conversely, the decoder on the right side serves as an expanding path that reconstructs spatial information to produce the final segmentation output. Spatial information reconstruction is achieved through upsampling operations at each decoding level. During upsampling, a 2 × 2 transposed convolution is used to double the spatial dimensions of the feature maps at each level, while halving the number of channels. The transposed feature maps are then concatenated with low-resolution feature maps from the corresponding encoder level. This concatenation allows the decoder to utilise information from both paths, facilitating accurate segmentation. The concatenated feature maps are then processed through the neural unit. This process is repeated until the final decoding level. In this study, zero-padding was applied to avoid cropping of feature maps during concatenation to prevent the loss of border pixels during convolutions in the encoder.

After the final basic block, a 1 × 1 convolution layer with two output channels is used to reduce the number of channels to match the number of classes for the binary segmentation task. Finally, a softmax activation function converts the output into a probability map, indicating the likelihood of each pixel belonging to the output classes.

3.6.2 Res-UNet

Res-UNet, also known as Deep Residual U-Net, is a powerful semantic segmentation model that incorporates residual connections into the U-Net architecture as proposed by Zhang, Liu and Wang (2018). While it retains the encoder-bridge-decoder structure of U-Net, it employs different basic building blocks. Instead of the plain unit used in U-Net, Res-UNet utilises pre-activated residual units. A typical residual unit contains the same layers as in a plain U-Net neural unit, but adds an identity mapping operation as shown in Figure 3.11. This identity mapping helps mitigate the vanishing gradient problem by offering a direct route for information flow through the network (He et al., 2016a). Specifically, a residual unit processes input *x,* produces *F(x)* through weighted layers, and generates output of *F(x)* + *x*, where the addition of the input *x* to the feature maps *F(x)* constitutes the identity mapping (often referred to as a residual connection).

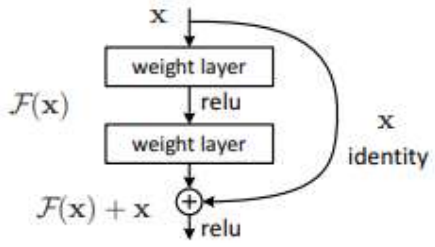


Figure 3.11: A residual unit (He et al., 2016a)**.**

In addition to identity mappings, Res-UNet employs fully pre-activated residual units, which have been shown to enhance model performance (He et al., 2016b). These units incorporate batch normalisation (BN) and ReLU activation

layers before convolution, a setup that pre-activates the convolution. Batch normalisation normalises the inputs to each layer, which can improve training stability and accelerate convergence (Goodfellow, Bengio and Courville, 2016). He et al. (2016b) demonstrated that this arrangement facilitates optimisation and results in lower classification errors. Moreover, batch normalisation in pre-activation also helps to regularise the models. An illustration of a full pre-activation residual unit is shown in Figure 3.12.

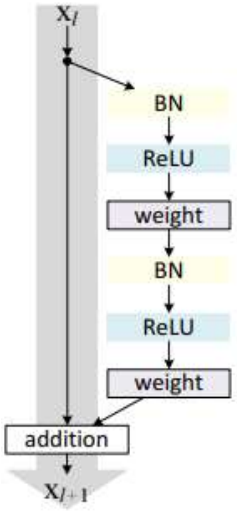


Figure 3.12: Full pre-activation residual unit (He et al., 2016b).

The architecture of Res-UNet used in this study is shown in Figure 3.13. Like U-Net, Res-UNet features three neural units in both the encoder and decoder connected by a bridge. However, in Res-UNet, downsampling is achieved by setting the stride to 2 in the first convolutional layer of each residual unit in the encoder, rather than using max-pooling as in U-Net. The processes of upsampling, concatenation, and segmentation output generation follow the same procedure as in the U-Net architecture.

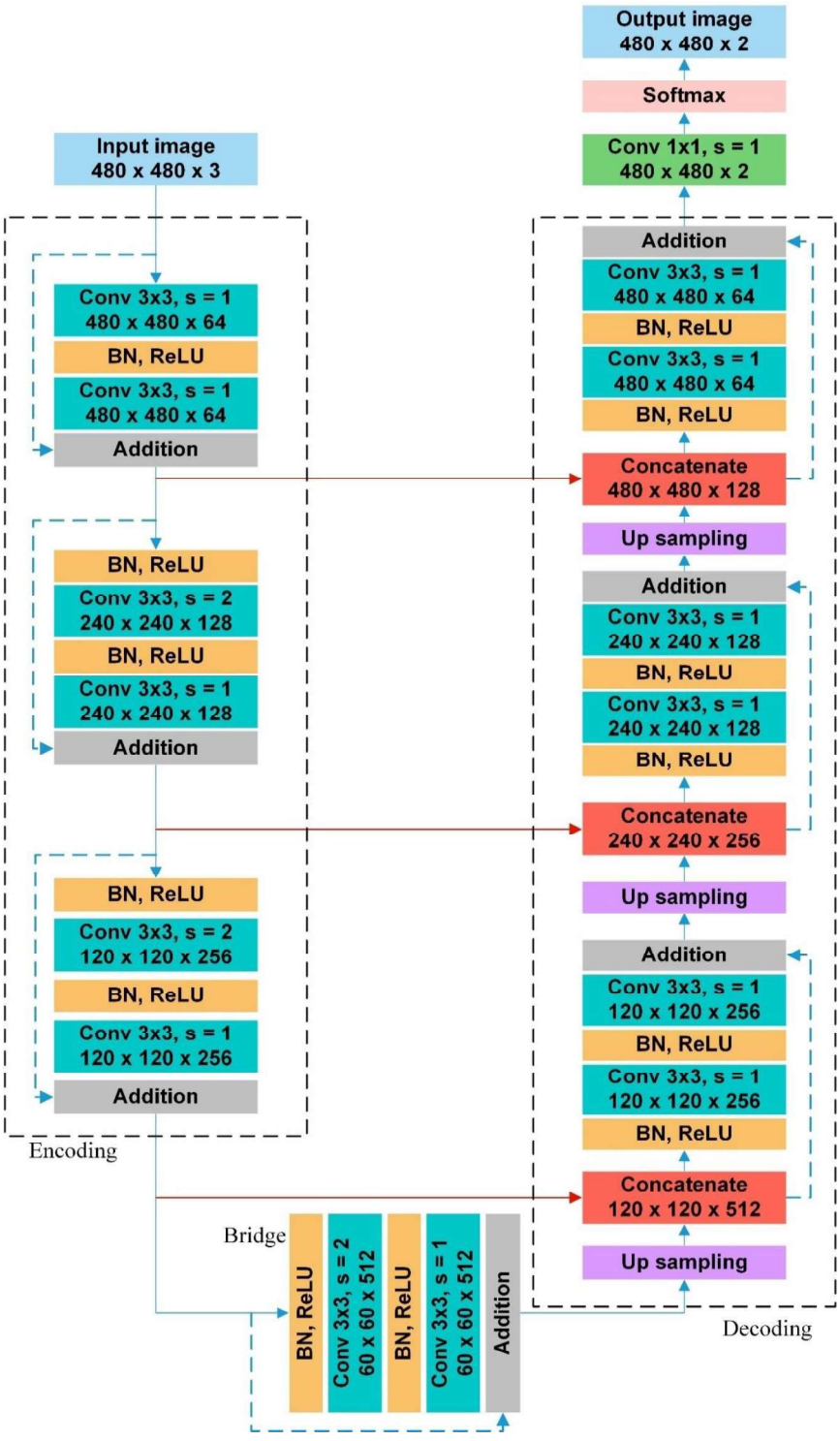


Figure 3.13: Res-UNet architecture. s represents stride. Output size of feature maps is annotated as height × width × number of channels.

3.6.3 DeepLabv3+

First introduced by Chen et al. (2018), DeepLabv3+ is a cutting-edge semantic segmentation model that leverages the combined strengths of spatial pyramid pooling and an encoder-decoder structure within deep neural networks. DeepLabv3+ encodes rich semantic information through its encoder, which includes a backbone network responsible for feature extraction and pooling operations, followed by the Atrous Spatial Pyramid Pooling (ASPP) module. To produce output, a simple decoder path is subsequently employed for reconstructing detailed object boundaries. The architecture of DeepLabv3+ is illustrated in Figure 3.14.

The backbone network typically comprises a deep convolutional neural network designed to extract high-level features for input into the ASPP module. It also transmits low-level features directly to the decoder path, facilitating their concatenation with the encoder's output to obtain more comprehensive representations. The training and performance of DeepLabv3+ benefit from the use of pre-trained backbone networks (Das et al., 2021). In this study, DeepLabv3+ models with four pre-trained CNN backbones, i.e., MobileNetV2, ResNet-18, ResNet-50, and Xception were evaluated for the task of soil desiccation crack segmentation. Descriptions of the backbones are provided in the subsections.

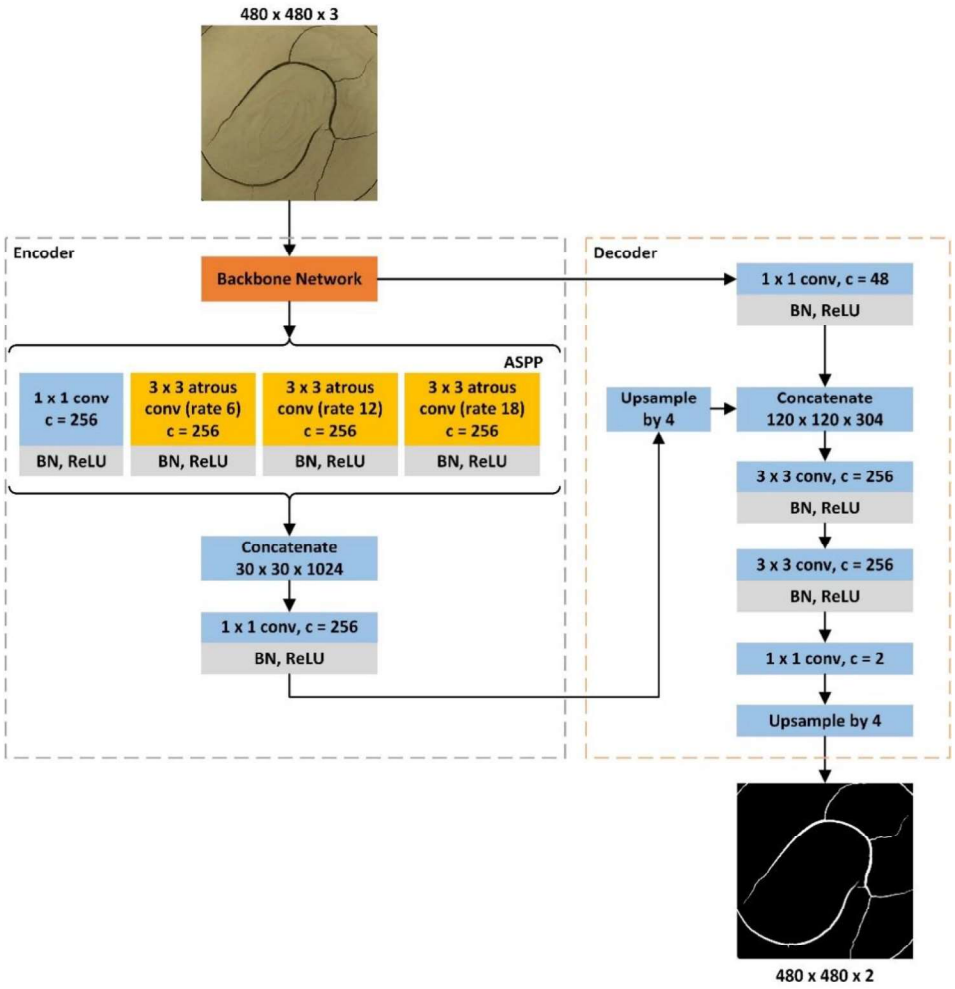


Figure 3.14: DeepLabv3+ model with an encoder-decoder structure.

The Atrous Spatial Pyramid Pooling (ASPP) module employs a series of parallel atrous convolutions and pooling operations to capture image information at multiple scales (Chen et al., 2017). In DeepLabv3+, ASPP integrates several key components, including atrous convolution, depthwise separable convolution, and spatial pyramid pooling (Chen et al., 2018). Atrous convolution is a variant of convolution that increases the receptive field without significantly raising the computational cost by using a dilated kernel. The kernel size is adjusted by dilation rate, which determines the spacing between kernel

weights. The concept of atrous convolution is illustrated in Figure 3.15. The dilation rate can be interpreted as the stride taken from one weight to another. For instance, with the rate of 1, the 3 × 3 kernel functions as a standard convolution. When the rate is increased to 2, the kernel weights are spaced apart, forming a 5 × 5 kernel with gaps, enabling the convolution to encode denser contextual information without extra computational effort.

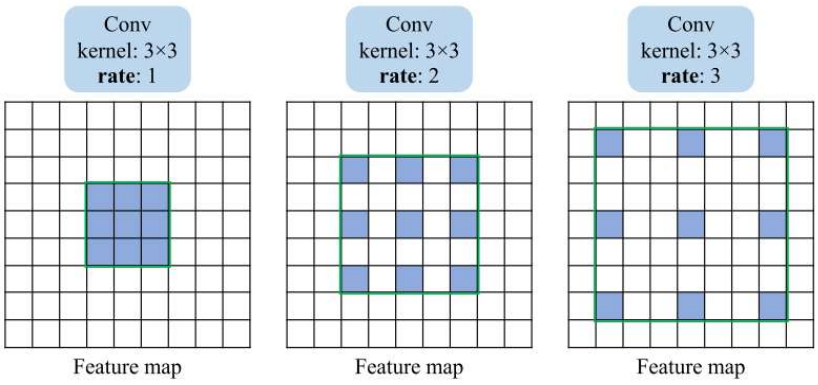


Figure 3.15: Atrous convolution at various atrous rates (Das et al., 2021)**.**

Chen et al. (2018) advanced atrous convolution by combining it with depthwise separable convolution to create atrous separable convolution. Depthwise separable convolution significantly reduces the computational cost of convolutional operations without degrading the model performance. It achieves this by decomposing the standard convolution into two components: depthwise convolution and pointwise convolution. Among them, depthwise convolution processes each input channel separately by applying individual filters, followed by pointwise convolution with 1 × 1 kernel to combine the outputs from the depthwise convolution. The number of output channels is adjusted by increasing the channels in pointwise convolution. Atrous separable

convolution incorporates atrous convolution kernels within the depthwise separable convolution framework, as depicted in Figure 3.16.

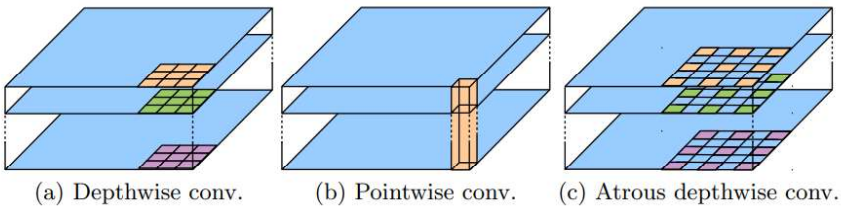


Figure 3.16: Demonstration of atrous separable convolution with a dilation rate of 2 (Chen et al., 2018)**.**

Spatial Pyramid Pooling (SPP) involves dividing an image into regions at various scales, performing pooling operations on each region, and concatenating the results into a fixed-length output, irrespective of the input image size (Grauman and Darrell, 2005; Lazebnik, Schmid and Ponce, 2006; He et al., 2015b). ASPP in DeepLabv3+ adopts the SPP concept by using atrous convolutions with different dilation rates, thereby enabling the network to capture multi-scale and rich contextual information from larger receptive fields.

In this study, the ASPP module comprised one 1 × 1 convolution, along with three 3 × 3 parallel convolutions with atrous rates of 6, 12, and 18 with a downsampling factor of 16. All convolutions featured 256 channels, followed by a BN layer and a ReLU layer. The image pooling branch as described by Chen et al. (2018) was omitted because the features extracted by the backbone networks already encapsulate image-level information. Although atrous separable convolutions were employed in ASPP, DeepLabv3+ with ResNet

backbones utilised standard atrous convolution without depthwise separation. This choice was made to leverage the ResNets' ability to learn richer feature representations without compromising efficiency. The results from the four parallel branches were concatenated to produce the ASPP output with 1024 filters. A subsequent 1 × 1 convolution reduced the number of filters to 256, followed by batch normalisation and ReLU activation to generate high-level feature output.

In the decoder path, the spatial dimensions of the encoded feature maps were initially upsampled by a factor of 4, followed by concatenation with low-level features from the backbone network, which had been passed through a 1 × 1 convolution for dimensionality reduction. The concatenated features were then refined through two 3 × 3 convolutional operations and a 1 × 1 convolution for final channel reduction. Eventually, the spatial dimensions were fully reconstructed through upsampling with a factor of 4, then finally a softmax activation was used to produce the final segmentation map.

3.6.3.1 MobileNetV2

MobileNetV2 is a computationally efficient, lightweight neural network that utilises depthwise separable convolutions. Its primary innovation lies in the integration of inverted residual blocks with a linear bottleneck structure as its fundamental building blocks (Sandler et al., 2018). A typical residual unit features a skip connection at both the input and output of the block, with a high number of channels at the skip connection compared to the

convolutional layers in between. Specifically, the number of channels is wide at the beginning of the block, narrow in the middle, and widens again at the end. In contrast, inverted residual blocks start with narrow input channels and end with narrow output channels. This configuration significantly reduces the number of parameters in the model, thereby enhancing the computational efficiency.

However, the reduced number of channels resulting from the skip connections can potentially impair the network performance. To address this issue, Sandler et al. (2018) introduced the concept of the linear bottleneck which employs a linear output from the final convolution in the building blocks. The structure of the basic building block is detailed in Table 3.2 and illustrated in Figure 3.17. Each block processes an input of size height × width × channel ($h \times w \times k$). Initially, a 1 × 1 convolution with a number of channels determined by the expansion factor (t) is applied to expand the channel dimensions to $t \times k$. This is followed by a 3 × 3 depthwise separable convolution with a stride of s for feature extraction. ReLU6 is used to introduce bounded non-linearity with an upper limit of 6. This capping improves training stability for low-precision arithmetic, which is often used in resource-constrained devices (Sandler et al., 2018). Eventually, a 1 × 1 convolution with k' output channels is applied to generate the block's output, which serves as input to the subsequent block. When the depthwise convolution has a stride of one, the input is combined with the output of the block through a skip connection.

Table 3.2: Linear bottleneck residual block structure (Sandler et al., 2018).

Input	Operator	Output
$h \times w \times k$	1 x 1 conv2d, BN, ReLU6	$h \times w \times (tk)$
$h \times w \times tk$	3 x 3 Depthwise s=s, BN, ReLU6	$h/s \times w/s \times (tk)$
$h/s \times w/s \times tk$	1 x 1 conv2d, BN	$h/s \times w/s \times k'$

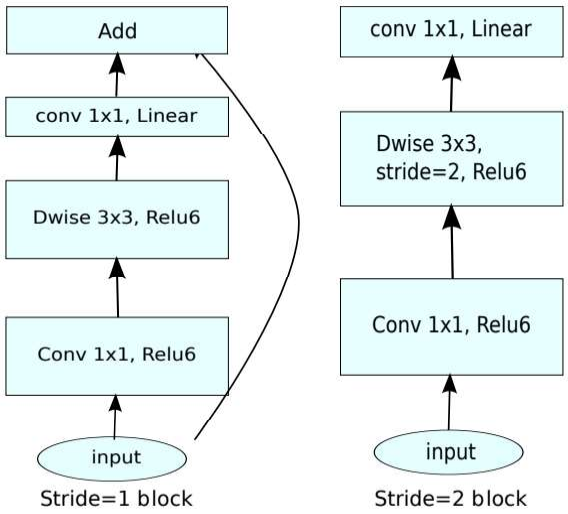


Figure 3.17: MobileNetV2 basic building block (Sandler et al., 2018).

The architecture of MobileNetV2 backbone for DeepLabv3+ is outlined in Table 3.3. As the encoder of DeepLabv3+, MobileNetV2 processes images with spatial resolution that adheres to a downsampling factor of 16 during feature extraction. The input image was initially processed with a standard 3 × 3 convolution with a stride of 2 to produce a reduced feature map with 32 filters. This is followed by a series of building blocks, referred to as bottleneck operators in the table. The table details the structure in sequence, where n indicates that the same operator with the same number of output

channels is repeated by n times. The building block configuration is determined based on the number of strides, as illustrated in Figure 3.17. Each sequence has a stride of s for the first block and a stride of 1 for all the subsequent blocks in the sequence. For example, at the input size of $240^2 \times 16$, the first bottleneck block follows a stride 2 configuration (without skip connection), and the second block is a stride 1 configuration with input from the first block. Both blocks have the same output channels of 24. Conversely, in sequences with dimensions $30^2 \times 64$ and $30^2 \times 96$, although the first bottleneck block has a stride of 1, these blocks do not include skip connections. As shown in Table 3.3, the MobileNetV2 backbone produces a $30^2 \times 320$ output tensor, which serves as the input for the ASPP module.

Table 3.3: MobileNetV2 architecture as DeepLabv3+ backbone.

Input	Operator	Expansion factor, t	Number of output channels, c	Module repetitions, n	Stride, s
$480^2 \times 3$	Conv2d	-	32	1	2
$240^2 \times 32$	bottleneck	1	16	1	1
$240^2 \times 16$	bottleneck	6	24	2	2
$120^2 \times 24$	bottleneck	6	32	3	2
$60^2 \times 32$	bottleneck	6	64	4	2
$30^2 \times 64$	bottleneck	6	96	3	1
$30^2 \times 96$	bottleneck	6	160	3	1
$30^2 \times 160$	bottleneck	6	320	1	1

3.6.3.2 ResNet-18 and ResNet-50

ResNet, or Residual Network, is a groundbreaking neural network architecture that enabled the training of extremely deep networks without facing the vanishing gradient problem (He et al., 2016a). This architecture introduced residual connections, a form of skip connection that bypasses certain layers and performs identity mapping, as exemplified in the Res-UNet architecture. ResNet-18 and ResNet-50 are variants of the ResNet architecture, with the numbers 18 and 50 representing the respective number of layers. Both ResNet-18 and ResNet-50 have 1 fully connected layer, with ResNet-18 having 17 convolutional layers and ResNet-50 having 49 convolutional layers. When used as backbones for DeepLabv3+, the fully connected layer is omitted, and the last convolutional layer is connected to the ASPP module.

The architectures of the residual networks employed are detailed in Table 3.4. According to the table, the building blocks of ResNet-18 consist of two 3 × 3 convolutions with specified channels, each followed by a BN and ReLU activation, with the number of blocks stacked indicated. The residual connections are implemented as illustrated in Figure 3.11. The ReLU activation is applied after the identity mapping and addition operation in the last convolution. ResNet-50 adheres to the same underlying concept but features a different building block and architecture configuration, as listed in the table. Notably, the first convolutional layers in conv3_x and conv4_x stages use a stride of two to achieve downsampling.

Table 3.4: Architecture of ResNet backbones.

Layer Name	Output size	18-layer	50-layer
conv1	240×240	7×7, 64, stride 2	
	120×120	3×3 max pool, stride 2	
conv2_x	120×120	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	60×60	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$
conv4_x	30×30	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$
conv5_x	30×30	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$

3.6.3.3 Xception

Extreme Inception, or Xception, is a CNN architecture that enhances the Inception model by substituting standard convolutions with depthwise separable convolutions (Chollet, 2017). While the Inception model is known for its effective use of separate convolutions to capture cross-channel and spatial correlations independently before concatenating the results, Chollet (2017) enhanced this approach by employing depthwise separable convolution. This modification not only aligns with the extreme version of the Inception V3 but is also significantly easier to model.

The Xception backbone architecture used in this study is illustrated in Figure 3.18. The backbone consists of 36 convolutional layers organised into 14 units, with the first and last units lacking residual connections. Each 'SeparableConv' layer comprises a 3 × 3 depthwise convolution operation, followed by a 1 × 1 pointwise convolution operation, with a channel expansion factor of 1. Batch normalisation is applied after each convolutional layer, as depicted in the figure. The input image undergoes initial processing through the entry flow, which uses a downsampling factor of 16. This is followed by eight middle modules for high-level feature extraction and concludes with the end flow, which provides the input to the ASPP module.

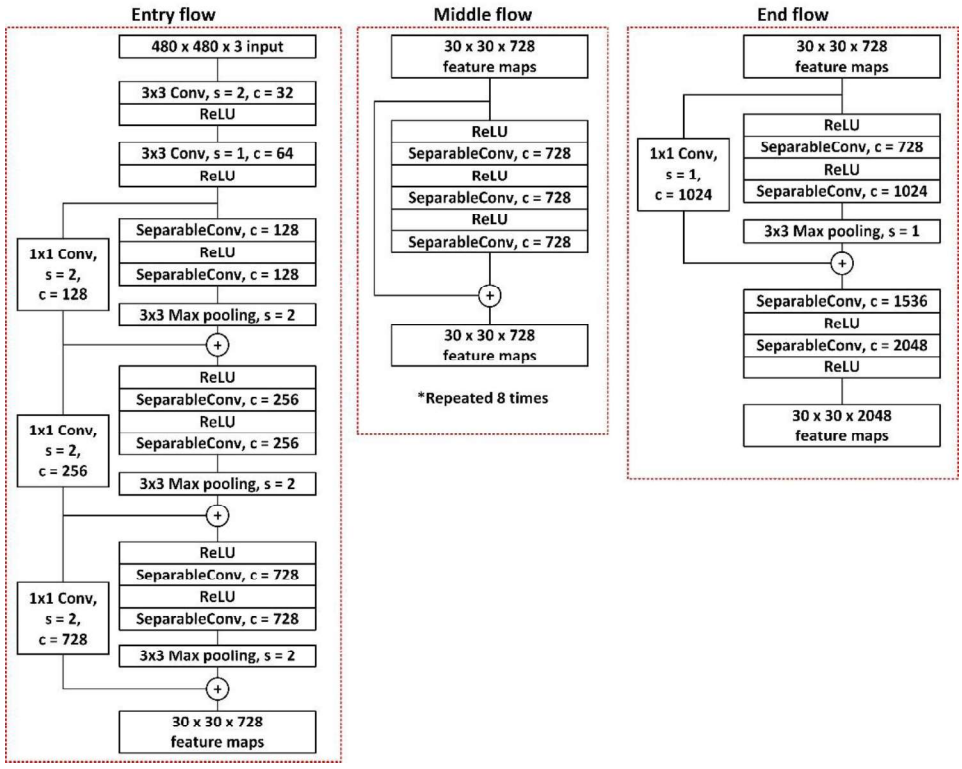


Figure 3.18: Architecture of Xception backbone. s and c stand for stride and number of output channels.

3.7 Model Training and Validation

Training deep neural networks involves a sequence of interconnected processes aimed at optimising the model's generalisation ability. The training process involved the repetition of weights and biases tuning in the network by passing the training images to it. This section outlines the methodology employed to train the deep learning algorithms. The MATLAB Deep Learning Toolbox was used to implement all the deep learning models. Training was conducted on an Nvidia RTX 3060 GPU with 12 GB of memory.

3.7.1 Weights and Biases

In a convolutional neural network (CNN), weights and biases are the fundamental parameters that the network learns during training. Weights are the parameters associated with the connections between different layers' neurons (Goodfellow, Bengio and Courville, 2016). In the context of CNNs, weights are the filters (also known as kernels) that are applied to the input data. Each filter moves across the input image, performing a convolution operation to produce feature maps. These feature maps emphasise different aspects of the input, such as edges, textures, and other patterns. The values of the weights determine the strength and orientation of these features, essentially acting as pattern detectors within the image.

Biases are additional parameters in the neural network that allow the model to have greater flexibility and to fit the training data more accurately

(Goodfellow, Bengio and Courville, 2016). Each neuron is associated with a bias value, which is added to the weighted sum of its inputs prior to the ensuing activation function. The purpose of the bias is to shift the activation function, which helps the model to better capture the relationship between input and output.

In a CNN, the convolution operation involves sliding the filter (weights) across the input image and calculating the dot product between the filter and its covered sections of the input. This process is repeated across the entire image, generating a feature map that represents the specific features detected by the current filter. The bias is then added to the result of the convolution before passing it through an activation function, such as ReLU, for nonlinear output.

3.7.2 Initialisation

The training of neural networks aims to optimise the weights and biases in its layers through iterative processes that converge to an optimal solution. These parameters critically influence the network's ability to generalise to new and unseen data. Proper selection of initial values for weights and biases before optimisation is crucial, as it enhances convergence speed and stability in network performance. This practice is known as parameter initialisation (Goodfellow, Bengio and Courville, 2016).

For the trainable layers in Res-UNet and DeepLabv3+ architectures, Glorot (also known as Xavier) initialiser was employed for weight initialisation.

The Glorot initialiser initialises weights from a uniform distribution with a range where the variance of the weights is inversely proportional to the sum of the number of input and output units (Glorot and Bengio, 2010). Although originally designed for a linear activation function, the Glorot initialiser is effective with ReLU (non-linear activation function) when batch normalisation is applied (Goodfellow, Bengio and Courville, 2016). Batch normalisation helps to mitigate issues such as exploding or vanishing gradients, thereby reducing the impact of the initialisation strategy. Since U-Net architecture lacks a batch normalisation layer, the He initialiser was used instead (He et al., 2015a). The He initialiser adjusts the initial weights based on the number of inputs, which is suitable for ReLU activation.

In this study, transfer learning strategy was employed exclusively for DeepLabv3+ models, as MATLAB provides pre-trained network packages for the backbones used in constructing these models. Transfer learning involves using pre-trained weights, typically trained on large datasets, to initialise the current network. This approach reduces training effort and improves performance by leveraging knowledge gained from a larger dataset (Han et al., 2022). The trainable weights in MobileNetV2, ResNet-18, ResNet-50, and Xception backbones for DeepLabv3+ were initialised with pre-trained weights trained on over a million images containing 1000 object categories from ImageNet. In contrast, biases were initialised to zero to avoid introducing any bias towards specific output.

3.7.3 Forward Propagation

As training begins, the input data is propagated through the network in a feedforward manner. Various operations are applied to the data as it traverses through convolutional, activation functions, batch normalisation layers, concatenation operations, pooling layers, and fully connected layers. At each stage, features and representations are extracted from the input data, capturing different levels of detail and spatial information. Ultimately, the network learns to assemble these patterns, enabling it to make predictions based on the learned representation.

3.7.4 Loss Function

Loss function is a critical component in the training process, as it measures the discrepancy between the network's predictions and ground truth labels during forward passes (Goodfellow, Bengio and Courville, 2016). Deep learning algorithms aim to optimise the model by minimising the loss function. This is achieved by iteratively adjusting the network's parameters during training to converge on optimal values that minimise loss. By minimising the loss function, the network improves its ability to make predictions with minimal error, thereby enhancing performance and accuracy.

In this study, the focal loss function was used to train all the CNN algorithms. The selection of this loss function addresses the issue of class imbalance, which is prevalent in crack detection tasks. In crack image datasets,

background pixels typically outnumber crack pixels, resulting in a model that is biased towards the majority class during training (Fu et al., 2022). The focal loss function modifies the traditional cross-entropy loss function in order to mitigate class imbalance. It introduces a modulating factor that increases the emphasis on difficult-to-classify examples, thereby enabling the model to focus more on learning challenging instances (Lin et al., 2018). The focal loss function is defined in Equation (1).

$$\mathrm{FL}\left(p_t\right) = -\alpha_t \left(1 - p_t\right)^{\gamma} \log\left(p_t\right) \tag{1}$$

where p_t is the predicted probability, FL(p_t) is the focal loss for a given p_t, α_t is the balancing weight ranging from 0 to 1, and γ is the focusing parameter. A typical value for γ is 2, as it provides an effective focusing factor; however, values in the range of 0 to 5 can be adaptable based on experimentation (Lin et al., 2018). In this study, α_t and γ were set to 0.8 and 2, respectively. This configuration allows α_t to give more weight to the minority class when its value is greater than 0.5, and γ to enhance the down-weighting of majority class samples.

3.7.5 Backpropagation

The process of calculating the gradient of the loss function in relation to the network's parameters is known as backpropagation (Goodfellow, Bengio and Courville, 2016). During backpropagation, the chain rule of calculus is applied to propagate errors backwards through the network. Subsequently, an

optimiser uses these gradients to update the parameters, facilitating the optimisation of the algorithm.

3.7.6 Parameter Update

Due to the inherently sparse gradient in semantic segmentation task, the loss function often exhibits non-convex behaviour. In non-convex functions, the presence of obscure local minima and plateaus can challenge the convergence of neural networks (Hadinata et al., 2021). To address these challenges, Kingma and Ba (2015) proposed the Adam (adaptive moment estimation) optimiser. Adam optimiser works by computing adaptive learning rates for each parameter separately, enabling the network to effectively navigate through these obstacles. The formulas for parameter updates using Adam optimiser are outlined below in their execution order.

$$g_t = \nabla FL\left(\theta_t\right) \tag{2}$$

$$m_t = \beta_1 \cdot m_{t-1} + \left(1-\beta_1\right) \cdot g_t \tag{3}$$

$$v_t = \beta_2 \cdot v_{t-1} + \left(1-\beta_2\right) \cdot g_t^2 \tag{4}$$

$$m_t = \frac{m_t}{1-\beta_1^t} \tag{5}$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t} \tag{6}$$

$$\theta_{t+1} = \theta_t - \alpha \frac{m_t}{\sqrt{\hat{v}_t} + \varepsilon} \tag{7}$$

where g_t denotes the gradient vector at time step t, computed as the partial derivative of the loss function with respect to parameter θ_t. m_t and v_t represent the updates for first and second moment estimates, with β_1 (exponential decay rate for m_t) and β_2 (exponential decay rate for v_t) typically set to 0.9 and 0.999, respectively. The updates for moment estimates are initially biased and must be corrected, resulting in bias-corrected moment estimates denoted by $\hat{m}_t$ and $\hat{v}_t$. Finally, the parameters are updated as θ_{t+1}, where α is the step size (with a suggested value of 0.001), and ε is a small constant (set to 10^{-8}) to prevent division by zero (Kingma and Ba, 2015; Goodfellow, Bengio and Courville, 2016).

An initial learning rate of 0.0001 was used with the Adam optimiser, along with a decay factor of 0.1 that was applied at every 15 epochs to reduce the learning rate throughout the training process. Additionally, L2 regularisation was employed to mitigate overfitting. L2 regularisation (or ridge regression) reduces overfitting by adding a penalty to the coefficients (weights) in the loss function (Murphy, 2012). The L2 regularisation can be written as:

$$\text{L2 Regularisation} = \lambda \sum (w_i^2) \tag{8}$$

$$\text{Loss}_{\text{L2 regularised}} = \text{Loss}_{\text{original}} + \text{L2 Regularisation} \tag{9}$$

where λ is the regularisation factor controlling the strength of the regularisation, and w_i represents the coefficient. This regularisation term is then added to the original loss function, yielding the regularised loss function shown in Equation

(9). The regularisation factor was set to 0.0005 based on parameter experimentation.

3.7.7 Iteration

Considering computational resources, the image batch size was set to 8 per iteration, and all models were trained for a total of 50 epochs. An epoch equals a complete pass across the entire training dataset. To further mitigate overfitting, the training data was shuffled at the beginning of each epoch. Validation was conducted at every 40 iterations using validation dataset to facilitate hyperparameter tuning and performance monitoring.

3.8 Model Evaluation

This study employed various standards to assess the performance of deep learning models in the context of expansive soil desiccation crack segmentation. The evaluation encompasses several categories, including computational efforts, metric evaluation for a specific CNN, and geometrical parameters that describe crack networks.

3.8.1 Computational Performance

The evaluation of computational performance was conducted to assess the efficiency of the segmentation models. The measured aspects included training time, inference time, frames per second (FPS), and the number of

parameters. Training time refers to the total duration required to train the deep learning models using the training dataset; a longer training time indicates greater usage of computational resources. Inference time denotes the duration each model takes to process an image and produce a prediction or segmented mask, calculated by dividing the total inference time by the number of images processed. FPS measures how many frames per second the model is able to process during inference, with higher FPS indicating faster processing speed. The number of parameters represents the total amount of learnable weights and biases in a model, reflecting its complexity; a higher parameter count typically requires more computational resources. For real-time applications, lower inference times and higher FPS are preferred, as they signify faster processing speed.

3.8.2 Evaluation Metrics

All segmentation methods and models were evaluated using standard deep learning model evaluation metrics commonly employed in image segmentation tasks, such as precision, recall, F1 score, and IoU. The pixel-wise prediction results from the models were categorised into four groups: true positive (TP), which indicates crack pixels correctly identified; false positive (FP), where background pixels were incorrectly predicted as crack; true negative (TN) for correct background predictions; and false negative (FN), where crack pixels were incorrectly classified. Using these categorised prediction results, the metrics were computed according to the following equations:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{10}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{11}$$

$$\text{F1 score} = \frac{2 \times precision \times recall}{precision \times recall} \tag{12}$$

$$\text{IoU} = \frac{TP}{TP + FP + FN} \tag{13}$$

Precision and recall both measure the accuracy of positive predictions. Precision is defined as the proportion of correct positive predictions out of all positive predictions made, while recall measures the ratio of correct positive predictions to all actual positives (Pham, Ha and Kim, 2023). F1 score provides a single numerical value that summarises the model's overall performance by computing the harmonic mean of precision and recall. Lastly, IoU quantifies the overlapping regions between predicted and ground truth masks that assess the algorithm's ability to accurately localise the object.

In the context of image segmentation, IoU is often regarded as the most significant metric since it directly reflects how well the generated segmentation aligns with the true crack network. Precision reflects the model's ability to avoid false crack detection while recall demonstrates how sensitive the model is in ensuring all actual cracks are captured. F1 score balances precision and recall into a single measure. Altogether, these metrics provided a comprehensive evaluation of crack segmentation accuracy.

3.8.3 Geometrical Parameters

Geometrical parameters are crucial for quantitative analysis of desiccation cracks, as they support the study of cracking mechanisms, propagation, and engineering properties (Xu et al., 2022b). In this study, the accuracy of segmentation models was assessed using key parameters in crack network analysis. The geometric features were quantified from the segmentation masks using image processing techniques implemented through MATLAB's Image Processing Toolbox. The parameters considered, along with their algorithmic execution are described below:

1) Surface crack ratio, R_{SC}: This parameter measures the ratio of the total crack area at a given time to the soil's total surface area at time zero. *Algorithm Execution*: The total number of crack pixels (white pixels representing cracks) along with the total number of pixels in the soil image were counted. R_{SC} was computed by dividing the number of crack pixels by the total number of pixels.
2) Average crack width, w_{avg}: This parameter represents the mean width of cracks measured around the medial axis (skeleton) of the crack network. *Algorithm Execution*: (a) The Euclidean distance image was obtained by computing the Euclidean distance transform of the binary image. (b) The distance image was multiplied by the skeletonised binary image and two to obtain an image where pixel values represent crack diameters. (c) Crack widths were extracted by identifying non-zero pixels. (d) Mean crack width was computed by taking the mean value of the crack widths.

3) Maximum crack width, w_{max}: This parameter denotes the width of the widest crack measured around the medial axis of cracks. *Algorithm Execution*: This process is similar to that for the average crack width, but instead of calculating the mean, the maximum value of crack widths was determined.

4) Number of intersections, N_{int}: This parameter counts the number of points where cracks intersect each other. *Algorithm Execution*: (a) The skeletonised image was computed, and mini-branches were pruned with threshold specified as w_{max}. (b) The skeleton pixels were iterated by tracking 8-neighbours in a clockwise direction. Each black-to-white pixel change was counted as one transition (Liu et al., 2013). (c) Intersection points were recorded at the pixel point where the number of transitions is equal to 3.

5) Number of nodes, N_n: This parameter is the sum of the points where cracks intersect (intersections) and where cracks terminate (end nodes). *Algorithm Execution*: (a) Generate a binary image with only end notes using MATLAB built-in morphological operation to identify endpoints on the skeleton. (b) Count the number of end nodes and calculate N_n by adding the end nodes to N_{int}.

6) Number of segments, N_{seg}: This parameter counts the distinct crack segments, each running from one node to meet another. The number of crack segments is the count of distinct crack segments where each segment runs from one node to meet another one. *Algorithm Execution*: (a) MATLAB's 'bwlabel' function was used to identify and assign different labels to each connected component (segment) in the

skeletonised image. (b) N_{seg} was calculated by taking the absolute value of the subtraction of N_n from the total number of connected components.

7) Total crack length, L_{sum}: This parameter represents the cumulative length of all individual cracks detected. *Algorithm Execution*: Compute L_{sum} by counting all non-zero elements in the skeletonised binary image.
8) Average crack length, L_{avg}: This parameter denotes the mean length of all individual cracks detected. *Algorithm Execution*: Compute L_{avg} by dividing the L_{sum} by N_{seg}.

After quantifying the geometrical features of the crack network from the binary masks, the error rates for each parameter between the ground truth and model predictions were computed by Equation (14).

$$E = \frac{\left|\text{Prediction} - \text{Ground Truth}\right|}{\text{Ground Truth}} \times 100\% \tag{14}$$

3.8.4 Recognition Accuracy and Stability

With different methods excelling in different evaluation standards, the mean and standard deviation are utilised as indicators of accuracy and stability to determine the best performer across all methods. The formulas for each are expressed as:

$$\bar{x}(\%) = \frac{\sum x}{n} \tag{15}$$

$$\sigma(\%) = \sqrt{\frac{\sum\left(x_i - \overline{x}\right)^2}{n}} \tag{16}$$

where $\overline{x}$ represents the mean value, calculated as the sum of terms, x, divided by number of terms, n, and σ denotes the standard deviation. The mean and standard deviation for each method were computed across all evaluation metrics. For geometrical parameters, only R_{SC}, w_{avg}, w_{max}, N_{seg}, and L_{sum} are included in the mean and standard deviation computations. A higher mean in evaluation metrics indicates better performance, while a lower mean value in geometrical parameters indicates better performance. A lower standard deviation value signifies a more stable performance across all standards.

3.9 Summary

In summary, to achieve the stated objectives, laboratory tests were designed and conducted. Several state-of-the-art deep learning semantic segmentation models were employed for soil crack segmentation. The segmentation accuracy of each model was evaluated using standard metrics. Geometrical parameters that described the crack network were computed through quantification using image processing techniques.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

In this chapter, a detailed analysis of various crack detection and segmentation methods applied to soil crack images is presented. The primary focus is to evaluate the performance of these methods in terms of computational efficiency, segmentation accuracy, and error rates in the quantification of crack geometrical characteristics. All the metrics, standards, and parameters used for the evaluation are defined and explained in Section 3.8.

Section 4.2 presents the image-based soil desiccation tests conducted with the developed image acquisition tool setup for soil crack measurements. Section 4.3 consists of computational efficiency analysis for each crack segmentation method. Section 4.4 examines the overall detection and segmentation performance on the test dataset for each model using evaluation metrics. Section 4.5 presents an in-depth analysis of the model's accuracy in quantifying geometrical parameters of crack across the test dataset. Section 4.6 delves into the stability of the recognition and quantification performance by evaluating the models' performance across all standards. Section 4.7 investigates the case-wise segmentation performance of each model under various photographic conditions, showcasing the robustness and adaptability of

the methods. This section provides a comprehension of the strengths and limitations of each method in handling diverse real-world conditions.

4.2 Image-based Soil Desiccation Test

To obtain a diverse crack image dataset for the training of deep learning models, the prepared specimens were subjected to soil desiccation tests under various experimental conditions. A detailed breakdown of the experiment conditions simulated is provided in Table 4.1. For example, row 2 of the table indicates that two specimens, each 5 mm thick, were subjected to desiccation test consisting of three W-D cycles. The test conditions included a temperature of 45 ± 2 °C and relative humidity (RH) of 60 ± 5 %.

Table 4.1: Test conditions for different specimens.

Number of specimens (nos.)	Final settled thickness (mm)	Test conditions		
		Temperature (± 2 °C)	Humidity (± 5 %)	W-D cycle, D (nos)
1	5	45	40	4
2	5	45	60	3
1	10	45	40	4
2	10	45	60	3
2	10	55	40	3
2	10	105	0	3
2	10	105	0	4
2	15	45	60	3

The experiment was conducted with the setup shown in Figure 4.1. The setup provided two different experimental conditions, i.e. 45 ± 2 °C and 55 ± 2 °C. Each setup included a humidity chamber, two or four oven bulbs, a humidity and temperature data logger, and a camera. The humidity chamber is a thick glass enclosure with restricted airflow to minimise fluctuations in temperature and humidity inside the chamber. The oven bulbs regulated the testing temperatures, with two bulbs maintaining a temperature of 45 ± 2 °C while four bulbs achieved 55 ± 2 °C. The camera captured images of the soil surface crack networks throughout the test, providing the dataset for CNN algorithm training. The humidity and temperature data logger monitored the testing conditions throughout the process. The variations in relative humidity were controlled by placing the chamber in environments with different air circulation patterns. In rooms with fan airflow or open-air conditions, the relative humidity (RH) in the chamber remained 40 ± 5 %, whereas it was 60 ± 5 % in an air-conditioned room or an environment with enclosed air circulation. For the highest testing temperature of a constant 105 °C, oven drying procedures were used.

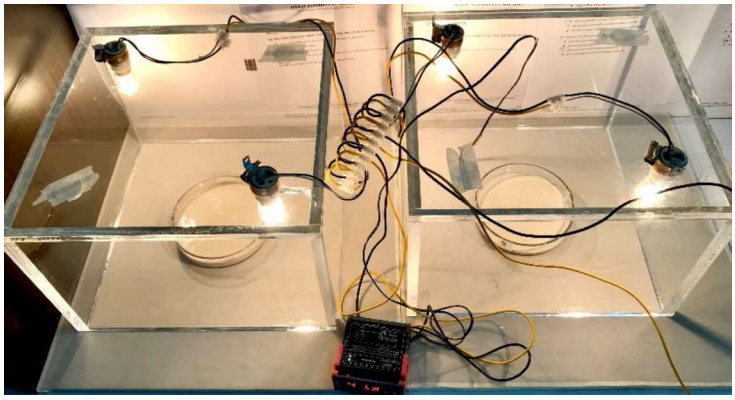


Figure 4.1: Experiment set-up for soil desiccation tests.

The simulation of desiccating temperatures of 45 ± 2 °C to 55 ± °C was based on several considerations from the literature on Malaysia's weather. On top of this is the fact that the highest temperature ever recorded in Malaysia was 40.1 °C in 1998 (Tan et al., 2021). They also found that Malaysia has experienced a significant warming trend, particularly across Peninsular Malaysia, indicating an upward shift in average temperatures. Driven by global warming, recent reports highlighted that heatwaves had led to higher peak temperatures recorded across the region, sometimes pushing temperatures beyond 40 °C (Koons, 2024). This evidence justified the use of 45 °C to simulate real soil desiccation conditions in Malaysia.

The 55 ± °C desiccating temperature is chosen to simulate even more extreme drying conditions that may become more frequent as climate change progresses. Global warming, coupled with regional phenomena like El Niño that bring extreme heatwaves across Southeast Asia, has caused temperatures to reach hazardous levels (*The Star*, 2023; Benjamin, 2024). As part of the region, Malaysia has experienced these extreme heat conditions, which are predicted to become more frequent due to the accelerating warming trend. The increasing intensity of heat patterns is likely to impact soil desiccation and cracking in Malaysia. Thus, the simulation of a higher desiccation temperature of 55 ± °C allows for a more comprehensive understanding of soil cracking behaviour under projected future conditions.

In natural environmental cyclic weathering conditions, expansive soils are subjected to repeated wetting and drying due to precipitation and evaporation, respectively (Al-Jeznawi, Sanchez and Al-Taie, 2020). To simulate these effects on expansive soil desiccation cracks, wetting-drying (W-D) cycles were introduced to the specimens, incorporating repeated wetting and drying phases throughout the experiment. The preparation of the slurry state specimen was considered the first wetting phase, while the subsequent desiccation process constituted the first drying phase. After each drying phase, the moisture lost during desiccation, measured as the specimen mass loss, was restored by adding an equivalent mass of distilled water (Tang et al., 2011a). Rehydrated specimens were allowed to sit for 24 hours during wetting paths. The drying phases were then conducted using the rehydrated specimens.

4.3 Computational Efficiency Analysis

Computational efficiency is a critical aspect in evaluating segmentation methods, as it provides insights into selecting the optimal method based on available resources and application constraints (Alom et al., 2019). The computational performance of various deep learning models and Otsu's method was analysed in terms of the model's number of parameters, training time, inference time, and frames per second (FPS). The first two parameters were not applicable to Otsu's segmentation method. A test dataset containing 20 images was used to measure the inference time and FPS for each method. The results are presented in Table 4.2.

Table 4.2: Computational performance for each segmentation method.

Method	Number of parameters ($\times 10^6$)	Training time (hr)	Total inference time (s)	Inference time per image (s)	Frames per second (FPS)
Otsu's	-	-	0.33	0.017	60.75
U-Net	7.69	1.67	4.25	0.212	4.71
Res-UNet	8.91	8.34	7.54	0.377	2.65
DeepLabv3+ (MobileNetV2)	6.78	0.88	2.99	0.150	6.68
DeepLabv3+ (ResNet-18)	20.61	0.75	2.63	0.132	7.60
DeepLabv3+ (ResNet-50)	43.98	1.28	4.25	0.213	4.70
DeepLabv3+ (Xception)	27.64	1.26	4.14	0.207	4.83

In terms of training resources, DeepLabv3+ with ResNet-50 had the highest number of parameters, exceeding the lowest count from DeepLabv3+ with MobileNetV2 by a factor of 6.5. On the other hand, Res-UNet and DeepLabv3+ with ResNet-18 exhibited the longest and shortest training times, at 8.34 hours and 0.75 hours respectively. The number of parameters in a model generally has a positive correlation with the model's ability to make accurate predictions but comes at the expense of increased memory consumption and operational cost (Alom et al., 2019). The high parameter count in DeepLabv3+

with ResNet-50 was primarily due to the numerous high-dimensional convolutional layers and skip connections in its backbone. In contrast, the use of depthwise separable convolution in DeepLabv3+ with MobileNetV2 backbone resulted in fewer parameters due to the reduced multiplication operations.

More parameters generally require more memory to store and may increase computational complexity during training, which potentially leads to longer training time. However, as shown in Table 4.2, DeepLabv3+ models (even those with a higher parameter count) all exhibited shorter training times compared to U-Net and Res-UNet. This is mainly due to the use of pre-trained backbones, which provided effective weight initialisation and training regularisation, resulting in faster convergence and improved generalisation (Gayakwad et al., 2021). Notably, Res-UNet took significantly longer to train compared to U-Net, despite sharing similar encoder-decoder depth. This discrepancy is likely attributed to the architectural differences that result in higher training complexity. In Res-UNet, residual connections and the 1×1 convolutions used at the residual branches, while alleviating vanishing gradient problems, also contribute to additional computational requirements. Each building block in Res-UNet, equipped with these features, necessitates greater forward pass and backward gradient computation efforts. Furthermore, unlike U-Net, Res-UNet employs batch normalisation layers to stabilise activation for each convolutional layer. Although this improves training stability and facilitates regularisation, BN layers impose additional memory overhead and can be computationally costly (Civitelli et al., 2023).

For inference time analysis, the total inference time to process 20 test images was recorded, and the inference time per image was calculated by dividing the total time by 20. It was unsurprising that Otsu's method had the shortest inference time, with a substantial discrepancy compared to the deep learning methods. This is because deep learning models involve complex neural network architectures with multiple operations, such as convolutions and skip connections, which require more computational effort for inference compared to the relatively straightforward computation in Otsu's algorithm. Res-UNet exhibited the longest inference time and the lowest FPS among all methods, which correlates with its high computational complexity during training, extending to the inference process.

A closer examination of Table 4.1 reveals a consistent trend among the Deeplabv3+ models: lighter backbones (MobileNetV2 and ResNet-18) achieved faster inference times and higher FPS, while deeper backbones (ResNet-50 and Xception) demonstrate moderate inference times and lower FPS. This suggests that among DeepLabv3+ models, deeper backbone architectures tend to perform worse, likely due to their increased computational complexity. Conversely, despite its lighter architecture, U-Net exhibited inference time and FPS values comparable to those of the DeepLabv3+ models with deeper backbones. These results suggest that U-Net did not benefit from its architecture and model size in terms of inference performance. Overall, DeepLabv3+ with lighter backbones, including MobileNetV2 and ResNet-18

achieved the best inference performance among all the deep learning models, with ResNet-18 obtaining the fastest inference time and highest FPS.

4.4 Crack Detection and Segmentation Performance

In this section, the crack detection performance for each segmentation method was analysed using evaluation criteria such as precision, recall, F1 Score, and IoU. The reported metrics are the average values computed over 20 images from the test dataset, thus representing the overall performance of each model in percentages. The IoU scores pertain specifically to foreground pixels (cracks) and do not include the mean IoU of both classes. This approach was chosen because, in binary segmentation tasks such as crack segmentations, the background pixels often dominate, leading to excessively high classification accuracy due to their abundance. Therefore, the mean IoU may not accurately reflect the actual segmentation performance in this context.

Table 4.3 summarises the performance of the different segmentation methods on these evaluation metrics. The comparison includes the traditional thresholding method (Otsu's method) and several deep learning methods (U-Net, Res-UNet, DeepLabv3+ with various backbone networks). The results indicate that the traditional thresholding method was significantly outperformed by deep learning approaches in all metrics except recall, where it exhibited only minor drawbacks. Res-UNet achieved the highest precision (91.75 %) while DeepLabv3+ with ResNet-50 excelled in the other metrics, with a recall of 92.30 %, F1 score of 91.47 %, and IoU of 84.29 %. These results suggested that

Res-UNet had the lowest false positive rate, making fewer mistakes when predicting pixels as cracks. However, DeepLabv3+ with ResNet-50 demonstrated greater sensitivity in detecting cracks, effectively identifying most actual cracks and achieving an excellent balance between precision and recall, as shown by its high F1 score. The highest IoU from DeepLabv3+ with ResNet-50 further underscores its reliability in accurately identifying crack regions, with strong overlap between predicted and actual crack regions.

Table 4.3: Segmentation performance on evaluation metrics.

Method	Precision (%)	Recall (%)	F1 score (%)	IoU (%)
Otsu's	13.31	83.11	22.94	12.96
U-Net	60.59	84.07	70.43	54.36
Res-UNet	91.75	86.94	89.28	80.64
DeepLabv3+ (MobileNetV2)	91.05	90.93	90.99	83.47
DeepLabv3+ (ResNet-18)	91.21	91.37	91.29	83.97
DeepLabv3+ (ResNet-50)	90.66	92.30	91.47	84.29
DeepLabv3+ (Xception)	90.49	91.09	90.78	83.13

To provide a clearer overview of the performances, a visual comparison of the metrics for the segmentation methods is illustrated in Figure 4.2. The chart illustrates that, in terms of precision, F1 score, and IoU, Otsu's

method exhibits notably poorer performance compared to all deep learning approaches, with 47 % lower in precision and F1 score than U-Net, and 41 % lower in IoU than U-Net. In contrast, all methods achieved satisfactory results for recall, with an increasing trend from Otsu's method (83.11 %) to U-Net (84.07 %), Res-UNet (86.94 %), and DeepLabv3+ models (all exceeding 90 %).

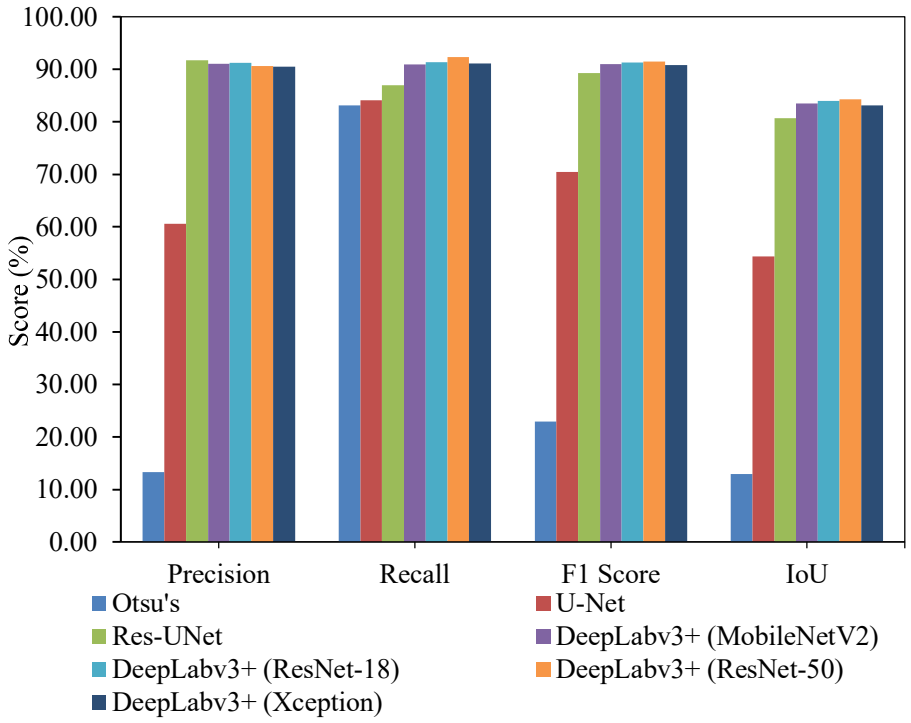


Figure 4.2: Metrics performance across different segmentation methods.

Otsu's method demonstrated a high recall of 83.11 %, indicating its ability to detect a significant portion of actual cracks. However, this high recall was accompanied by low precision (13.31 %) and low IoU (12.96 %). These metrics suggested that Otsu's method tended to overly identify pixels as cracks, resulting in a high false positive rate compared to actual crack instances. The method's tendency to make substantial positive inferences increased the

likelihood of true positive predictions and reduced the number of false negatives (cracks identified as background), thus leading to a high recall value. However, this approach diminished segmentation accuracy, as evidenced by the lower precision and IoU due to the increased number of false positives.

A closer examination of the metrics presented in Table 4.3 and Figure 4.2 reveals that among the deep learning methods, U-Net demonstrated moderate performance relative to the other models. U-Net achieved a precision of 60.59 % and an IoU of 54.36%, indicating its capability to recognise the crack detection tasks. Figure 4.3 provides an example of a segmentation mask generated by U-Net. It shows that while U-Net is effective at outlining the general crack network, it struggles to generalise effectively when the soil background has a slightly rougher texture, resulting in increased noise.

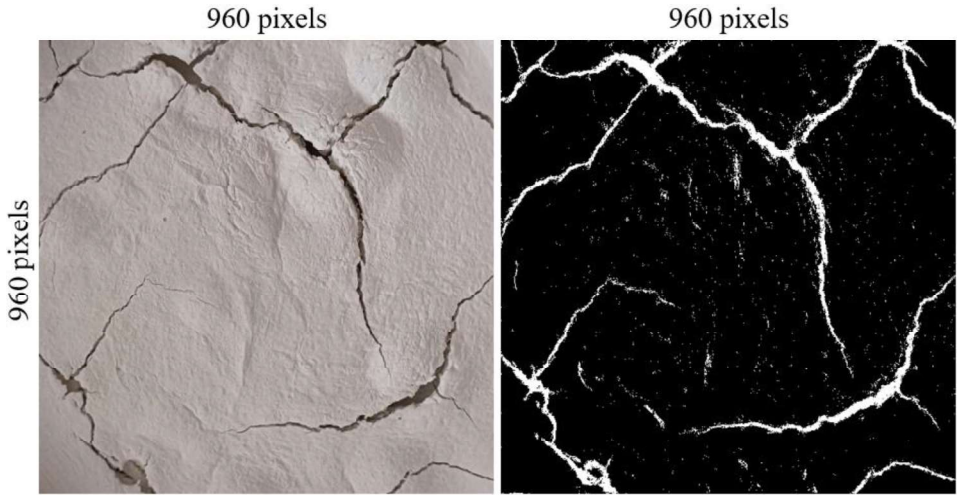


Figure 4.3: Original image (left) and U-Net segmentation mask (right).

The inclusion of residual connections and batch normalisation in Res-UNet led to significant improvements over U-Net model, with enhancements in

precision (31.16 % higher), F1 score (18.85 % higher), and IoU (26.28 % higher). However, as shown in section 4.3, these additional connections and batch normalisation layers considerably increased the computational complexity. In terms of precision, DeepLabv3+ models performed similarly to Res-UNet, with Res-UNet exceeding DeepLabv3+ by just 0.54 % to 1.26 %. For recall, the DeepLabv3+ models, with MobileNetV2 backbone achieving 90.93% and ResNet-50 backbone achieving 92.30%, outperformed Res-UNet (86.94%) by 3.99% and 5.36% respectively. Additionally, DeepLabv3+ models achieved IoU scores with percentage increases ranging from 2.49 % to 3.65 % over Res-UNet. Thus, it can be concluded that the overall performance of DeepLabv3+ models, irrespective of the chosen backbone networks, is superior among the segmentation methods evaluated.

Among the DeepLabv3+ models, it is observed that lighter backbones produced predictions with slightly better precision, with ResNet-18 (91.21 %) outperforming MobileNetV2 (91.05 %). All models demonstrated similar performance in terms of recall, with ResNet-50 leading, followed closely by ResNet-18, Xception, and MobileNetV2. Regarding IoU, ResNet-50 surpassed ResNet-18 by 0.32 %, MobileNetV2 by 0.82 %, and Xception by 1.16 % indicating its superior performance over lighter backbones. This suggests that deeper and more complex architectures can capture finer details in crack segmentation tasks.

Besides, it is evident that models equipped with standard convolutions, such as ResNet-18 and ResNet-50, outperformed those with depthwise

separable convolutions, including MobileNetV2 and Xception, in soil crack segmentation. Between MobileNetV2 and Xception, it is obvious that using depthwise separable convolution in the inverted residual building block with the linear bottleneck in MobileNetV2 is preferable to the approach used in Xception's residual blocks. MobileNetV2 demonstrated a significantly lower number of parameters and higher computational efficiency, while still achieving satisfactory overall segmentation performance.

In summary, DeepLabv3+ with ResNet-50 achieved the best segmentation accuracy according to its performance across evaluation metrics, though it required substantial computational resources. Among the lighter backbones, MobileNetV2 proved to be an excellent choice for applications with limited computing capacity, whereas ResNet-18 offered marginal improvements in segmentation accuracy with greater resources.

4.5 Crack Geometrical Characteristics Quantification Analysis

The quantification of geometrical features of the crack networks was conducted using the parameters defined in Section 3.8.3. The predicted masks generated through various segmentation methods were analysed, and the error rates against ground truth masks were computed. The average error percentages over 20 test images for each geometrical parameter and each method are presented in Table 4.4.

From the table, it is evident that traditional Otsu's segmentation method exhibited the highest error rates across all parameters, except for parameters like N_{int}, N_n, and N_{seg}. For N_{int}, N_n, and N_{seg}, U-Net performed the poorest, with errors of 50,023.13 %, 10,155.59 %, and 15,326.07 % respectively. The second-highest error rates were associated with Otsu's method with 23,474.74 %, 5,713.71 %, and 10,084.64 % error rates respectively. Among the deep learning methods, U-Net showed moderate performance across all parameters compared to the other methods, with particularly high errors in N_{int}, N_n, and N_{seg}. The integration of residual connections in Res-UNet led to a significant improvement in performance compared to U-Net, with substantially lower errors across all parameters. Notable improvements were observed in the accuracy of the number of intersections (approximately 14.2 times better), nodes (15.5 times better), crack segments (16.3 times better), and total crack length (12.7 times better) compared to U-Net.

The table indicates that all DeepLabv3+ variants exhibited significantly lower error rates in all parameters compared to Otsu's method and U-Net. The DeepLabv3+ variants also showed marked improvements in N_{int}, N_n, N_{seg}, and L_{sum}. Among the DeepLabv3+ models, the MobileNetV2 variant achieved the highest performance, with improvements of 66.4 times in N_{int}, 15.3 times in N_n, 29.3 times in N_{seg}, and 3.7 times in L_{sum} compared to Res-UNet. Conversely, DeepLabv3+ with ResNet-18 demonstrated the best performance in terms of surface crack ratio (7.04 %), average crack ratio (8.07 %), and maximum crack width (9.93 %). Other DeepLabv3+ variants also performed closely on these parameters, with ResNet-50 variant coming in second, followed

by MobileNetV2 and Xception. A closer inspection revealed that Xception variant had slightly higher error rates than the other DeepLabv3+ variants across all parameters, but still outperformed Otsu's, U-Net, and Res-UNet methods.

The results indicate that traditional methods, such as Otsu's global thresholding segmentation method, are inadequate for the segmentation of soil crack networks due to their high error rates in geometrical parameter quantification. In contrast, deep learning methods, especially DeepLabv3+ models, demonstrated superior performance in handling complex soil crack segmentation tasks, with significantly lower error rates compared to traditional methods.

Despite having substantially lower error rates in surface crack ratio and crack width computation compared to traditional methods, U-Net still showed considerable errors in detecting crack segments, as shown by its high error of 15,326.07 % in N_{seg}. This may be attributed largely to the noise present in its segmentation results, as illustrated in Figure 4.3. Conversely, the residual connections and batch normalisation in Res-UNet successfully enhanced the accuracy of the model close to a competitive level with DeepLabv3+ models, though its performance was limited to surface crack ratio (9.93 %) and crack width detection (around 14 % for w_{avg} and w_{max}). The high error rates in the number of intersections (3,300.71 %), nodes (614.84 %), and crack segments (886.16 %) suggest that Res-UNet is less effective in preserving the connectivity of the crack networks compared to DeepLabv3+ with error rates lower than 100 % in N_{seg}.

Among the deep learning methods, DeepLabv3+ with MobileNetV2 can be considered the most accurate due to its consistently low error rates across all parameters. Coupled with its computational efficiency, the model is well-suited for real-time and resource-constrained applications, providing satisfactory segmentation performance. The ResNet-18 variant offers another reliable option for soil crack segmentation tasks, with strong performance in surface crack ratio and crack width quantifications.

Table 4.4: Segmentation error on geometrical parameters by different methods.

Method	Error over 20 images, E (%)							
	R_{SC}	w_{avg}	w_{max}	N_{int}	N_n	N_{seg}	L_{sum}	L_{avg}
Otsu's	1029.42	209.16	754.04	23474.74	5713.71	10084.67	389.82	77.14
U-Net	46.31	47.67	19.53	50023.13	10155.59	15326.07	206.77	97.06
Res-UNet	9.93	13.52	14.22	3300.71	614.84	886.16	15.04	69.88
DeepLabv3+ (MobileNetV2)	7.89	9.52	10.21	48.94	37.63	29.24	3.20	25.75
DeepLabv3+ (ResNet-18)	7.04	8.07	9.93	102.30	50.10	52.63	3.61	31.38
DeepLabv3+ (ResNet-50)	7.54	9.03	10.28	110.46	42.72	47.61	3.31	29.73
DeepLabv3+ (Xception)	7.96	8.37	11.17	180.22	63.19	70.68	4.58	34.03

4.6 Accuracy and Stability of Recognition Performance

The accuracy and stability of recognition and quantification performance for different segmentation methods were evaluated using mean values ($\overline{x}$) and standard deviations (σ) for evaluation metrics and geometrical parameter errors. The results are presented in Table 4.5.

Table 4.5: Accuracy and stability of each method across various standards.

Methods	Evaluation Metrics		Geometrical Parameters	
	$\overline{x}$ (%)	σ (%)	$\overline{x}$ (%)	σ (%)
Otsu's	33.08	29.16	2493.42	3806.28
U-Net	67.36	11.22	3129.27	6098.76
Res-UNet	87.15	4.13	187.77	349.20
DeepLabv3+ (MobileNetV2)	89.11	3.26	12.01	8.95
DeepLabv3+ (ResNet-18)	89.46	3.17	16.25	18.30
DeepLabv3+ (ResNet-50)	89.68	3.17	15.55	16.20
DeepLabv3+ (Xception)	88.87	3.32	20.55	25.15

The evaluation metrics, which include precision, recall, F1 score, and IoU, reflect the accuracy of detection and segmentation. The traditional Otsu's segmentation method exhibited the lowest mean value (33.08%) and the highest

standard deviation (29.16%) in this category, indicating poor performance and high variability in crack detection. U-Net achieved a mean value of 67.36% with a standard deviation of 11.22%, representing a significant improvement over Otsu's method but still lacking behind other deep learning methods. DeepLabv3+ with ResNet-50 had the highest mean value at 89.68%, closely followed by ResNet-18 variant at 89.46%, both achieving the lowest standard deviation of 3.17%. This suggests the most excellent and consistent performance among all methods. DeepLabv3+ with MobileNetV2 and Xception also demonstrated comparable performance at both metrics against the ResNets variants, indicating consistently high performance among DeepLabv3+ models.

Regarding the quantification of crack geometrical parameters, U-Net was the poorest performer, with an exceptionally high mean value of 3,129.27% and a standard deviation of 6,098.76%, indicating significant inaccuracies and variability in geometrical quantification. Otsu's method was the second poorest performer, with a similarly high mean value and standard deviation for quantification errors. In contrast, Res-UNet, with its more advanced architecture, demonstrated significant improvement, with mean value and standard deviation reduced to 187.77% and 349.20%, respectively. The results indicate that DeepLabv3+ variants outperformed the other methods, showing superior performance in both mean value and standard deviation compared to Res-UNet, where DeepLabv3+ achieved 11 times and 20 times lower mean and standard deviation respectively. Among the DeepLabv3+ models, MobileNetV2 backbone achieved the lowest mean error of 12.01% and standard deviation of

8.95%, which were considerably lower than those of other variants, indicating exceptional accuracy and stability in crack network quantification.

The analysis of mean values and standard deviations underscores the superiority of deep learning-based models, particularly the DeepLabv3+ variants, over traditional segmentation methods in crack recognition and quantification. DeepLabv3+ models demonstrated remarkable stability and reliability in both recognition and quantification tasks, with higher accuracy and lower variability across various standards. Overall, DeepLabv3+ with MobileNetV2 is suggested as the best model due to its superior performance in geometrical parameter quantification and sufficiently high accuracy in evaluation metrics.

4.7 Case-wise Segmentation Performance

To further investigate the effectiveness and accuracy of the segmentation methods, the detection performance on images under various conditions was examined. The segmentation performances for each condition are presented in the following subsections, with the segmented masks generated by different methods illustrated. Cases A to H represent eight distinct crack image conditions, characterised by variations in the clarity of crack networks and edges, illuminations, image exposures, and other factors.

4.7.1 Case A: Standard Crack Network with Clear Definition

Case A represents a simple condition where the image features a crack network with distinct lines and edges under standard illumination and exposure settings. Figure 4.4 illustrates the image along with the corresponding masks generated by various segmentation methods, and their segmentation performance is summarised in Table 4.6. Visual observation of the segmented masks in Figure 4.4 indicates that all methods performed well under these conditions as the detected crack networks were sufficiently close to the ground truth. However, some noises, in the form of white speckles, are visible in the masks generated by U-Net and Otsu's method, with U-Net exhibiting a higher number of noises. The masks generated by Res-UNet and DeepLabv3+ variants closely match the ground truth, suggesting better segmentation capability under standard conditions.

From the top half of Table 4.6 it is evident that DeepLabv3+ with ResNet-18 achieved the highest precision (96.84 %), followed closely by Res-UNet (95.34 %) and DeepLabv3+ with MobileNetV2 (95.55 %). In terms of the ability to identify all relevant instances, DeepLabv3+ with ResNet-50 achieved the highest recall (97.02 %). Res-UNet and the DeepLabv3+ variants demonstrated a good balance between precision and recall, as indicated by their high F1 scores. Among the deep learning methods, IoU was the highest for Res-UNet (92.45 %), with DeepLabv3+ variants also performing well, all exceeding 91 %, reflecting a good overlap with the ground truth. Notably, Otsu's method exhibited superior performance by achieving the highest F1 score and IoU

among all methods. Its performance metrics surpassed U-Net by 16.5 % in precision, 11 % in F1 score, and 18.6 % in IoU. U-Net is the poorest performer across all evaluation metrics.

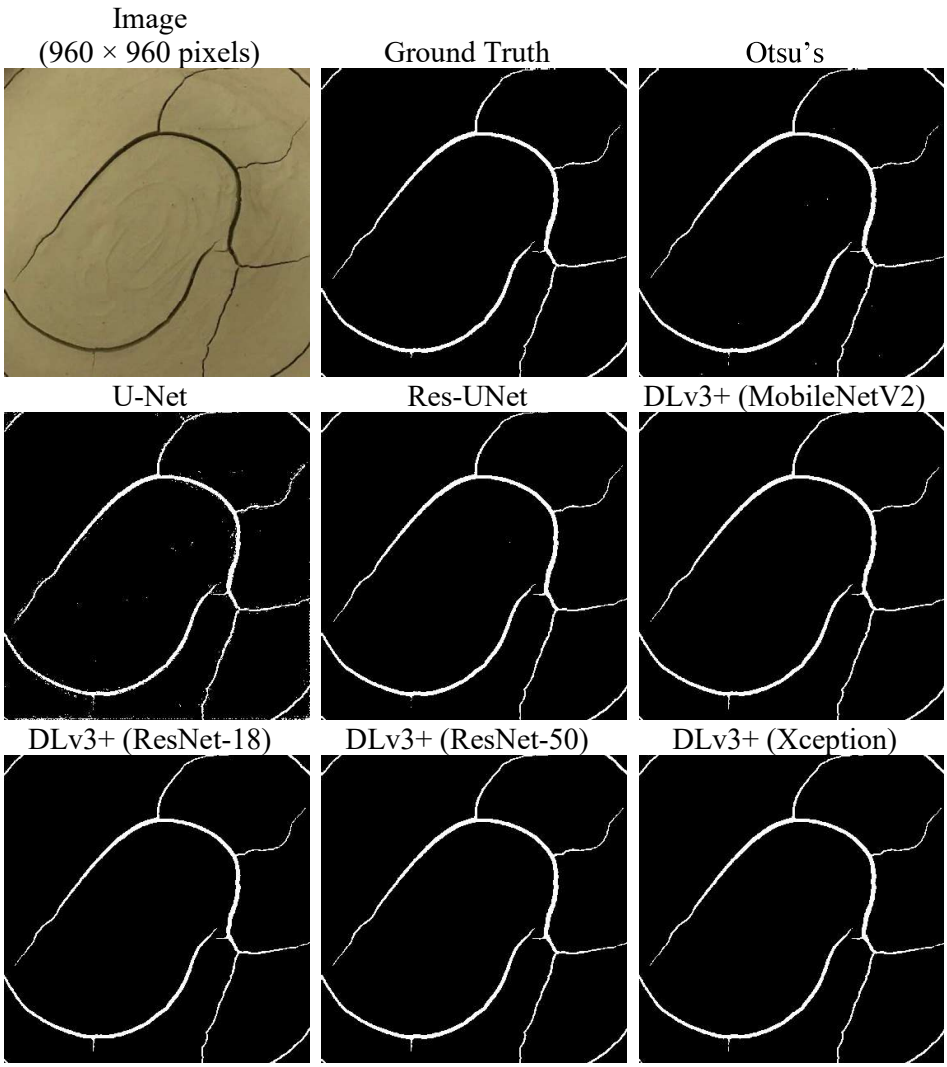


Figure 4.4: Segmentation visualisation for Case A.

The bottom half of the table presents the geometrical parameter error rates for the case. DeepLabv3+ with MobileNetV2 exhibited the lowest error in surface crack ratio detection (0.23 %), followed closely by the other methods. Conversely, Otsu's method and DeepLabv3+ with ResNet-18 led in w_{avg}, with

the latter having a 0.5 % higher error than the former. The remaining methods also showed low errors in w_{avg}, ranging from 2.78 % to 3.79 %. U-Net was the worst performer in R_{SC} (16.08 %) and w_{avg} (32.71 %) with considerably higher errors compared to the other methods. For maximum crack width (w_{max}), all methods performed excellently, with the highest error capped at 3.61 %. Notably, U-Net and DeepLabv3+ with ResNet-50 achieved zero errors. DeepLabv3+ variants excelled in detecting the N_{int}, N_n, and N_{seg}, with zero errors, indicating precise detection of connectivity in the crack network. Res-UNet came second with errors ranging from 6 % to 20 % errors, followed by Otsu's at 36 % to 80 %, and U-Net with soaring errors ranging from 3,500 % to 15,000 %. For crack length, Otsu's method had the lowest error in identifying total crack length, while DeepLabv3+ with ResNet-50 showed the lowest error in average crack length. All deep learning models performed well with low errors in both L_{sum} and L_{avg} except for U-Net, which exhibited nearly 100 % errors.

The results from Case A illustrate that deep learning methods, particularly those based on DeepLabv3+ architecture, outperformed traditional methods and simpler CNN like U-Net in segmenting crack networks under standard conditions. Despite having relatively low error rates across parameters such as R_{SC}, w_{avg}, w_{max}, and L_{sum}, Otsu's segmentation method is unsuitable for precise crack network recognition due to poor performance in identifying crack segments. Accurate identification of crack segments is important as it reflects the stage of soil disintegration; a higher number of crack segments usually indicates a more weathered and fragmented soil structure with reduced stability

(Tang et al., 2008). For this reason, Res-UNet is less effective compared to the DeepLabv3+ variants, which correctly identified the number of crack segments. The significant errors exhibited by U-Net may be attributed to extensive noise in its segmented mask, highlighting its limitation for complex segmentation tasks such as soil crack network recognition. DeepLabv3+ models, especially those with ResNet-18 and MobileNetV2 backbones, proved to be the most effective segmentation method for Case A, demonstrating high accuracy across all measured standards and low error rates in geometrical parameter computation, underscoring their robustness in handling the soil crack segmentation task.

Table 4.6: Case A segmentation performance.

Method	Otsu's	U-Net	Res-UNet	DeepLabv3+ (MobileNetV2)	DeepLabv3+ (ResNet-18)	DeepLabv3+ (ResNet-50)	DeepLabv3+ (Xception)
Evaluation Metrics (%):							
Precision	95.95	79.50	95.34	95.55	96.84	94.84	95.16
Recall	96.93	92.28	96.82	95.77	93.86	97.02	96.22
F1 Score	96.44	85.42	96.07	95.66	95.33	95.92	95.69
IoU	93.12	74.54	92.45	91.69	91.07	92.16	91.74
Error on Geometrical Parameters (%):							
R_{SC}	1.01	16.08	1.56	0.23	3.08	2.30	1.11
w_{avg}	0.40	32.71	2.96	2.78	0.94	3.79	3.58
w_{max}	3.61	0.00	3.61	2.23	2.23	0.00	2.90
N_{int}	80.00	15180.00	20.00	0.00	0.00	0.00	0.00
N_n	36.36	3509.09	9.09	0.00	0.00	0.00	0.00
N_{seg}	37.50	4662.50	6.25	0.00	0.00	0.00	0.00
L_{sum}	0.38	81.87	3.54	2.46	1.96	1.17	2.92
L_{avg}	27.00	96.18	2.89	2.46	1.96	1.17	2.92

4.7.2 Case B: Overexposure with Excessive Illumination

This case exemplifies a scenario involving overexposure issues, specifically excessive illumination on the soil surface. Figure 4.5 illustrates the image of Case B along with the segmentation masks generated by various methods. Visual inspection of these masks reveals that traditional method performs adequately in recognising the general crack network, albeit with some noise (present as white speckles). It can be concluded that overexposure did not significantly impact the performance of the simple Otsu's global thresholding segmentation. Conversely, while all the deep learning methods effectively generalise the crack network with minimal noise, U-Net model exhibited a notable increase in noise under overexposure conditions compared to the standard condition observed in Case A.

Res-UNet model, which incorporates residual connections and batch normalisation, substantially moderated the noise issue but did not perform as well as the DeepLabv3+ variants, which contained some non-crack speckles in the central region. Among the DeepLabv3+ models, those with ResNet-18 and ResNet-50 backbones showed superior alignment with the ground truth, with ResNet-50 performing the best. The crack network generated by DeepLabv3+ with ResNet-50 backbone exhibited improved connectivity and more effective preservation of the connections and intersections between crack lines.

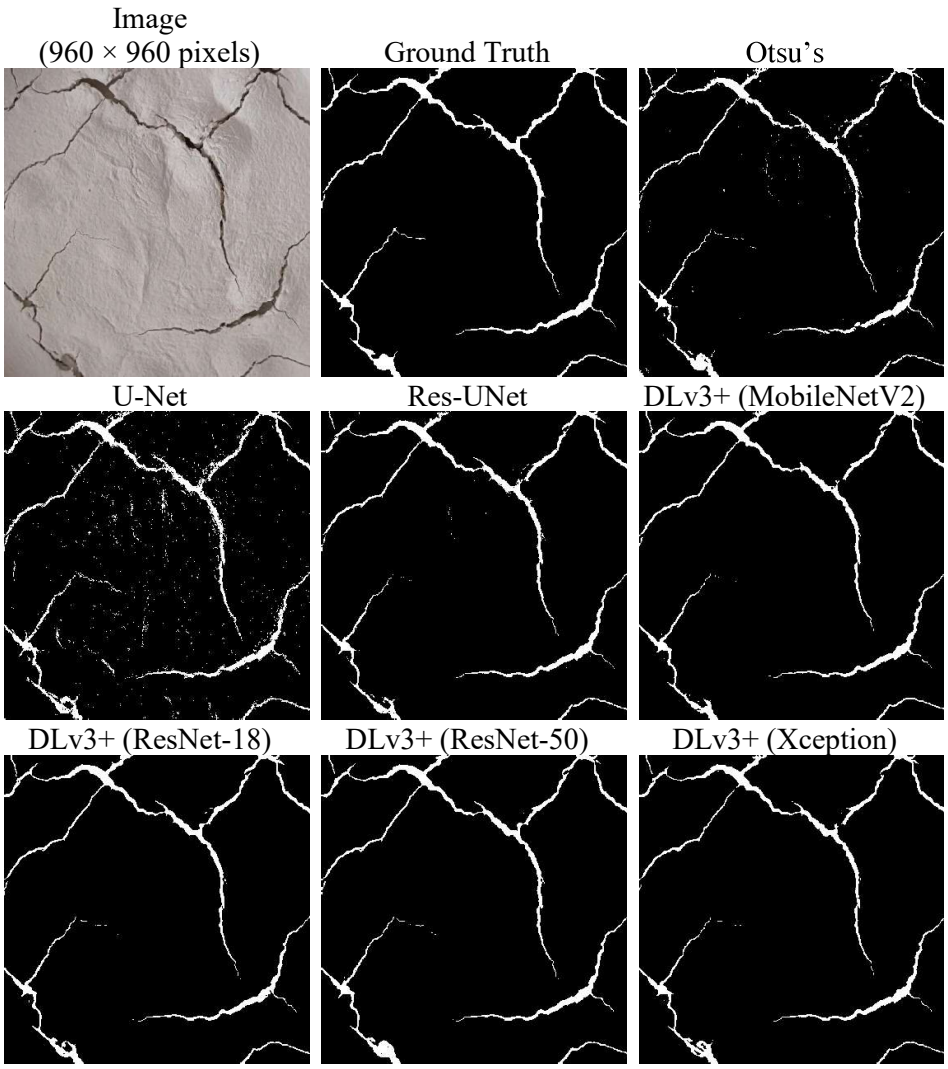


Figure 4.5: Segmentation visualisation for Case B.

Table 4.7 summarises the segmentation performance of various methods under the Case B condition. The results, as depicted in the table, are consistent with the visual observations previously described, with DeepLabv3+ variants outperforming other methods and U-Net ranking lowest. Specifically, DeepLabv3+ with Xception achieved the highest precision at 93.75 %, followed closely by other DeepLabv3+ variants, demonstrating the architecture's superior capability in identifying relevant crack pixels despite overexposure. In terms of recall, DeepLabv3+ with ResNet-50 led with 93.60 %, while other

methods fell behind, with percentage differences ranging from 3.2 % (ResNet-18) to 6.63 % (U-Net). The performance in the F1 score was similar to recall results, with DeepLabv3+ with ResNet-50 leading with a score of 93.25 %, while U-Net was the poorest performer with a score of 74.96 %. Additionally, DeepLabv3+ with ResNet-50 demonstrated the best overlap with ground truth under overexposure conditions, achieving an IoU of 87.36 %. All the other methods achieved IoU values above 81 %, except for U-Net, which had an IoU of 59.95 %. Compared to the evaluation metrics from Case A, there was a general decline in performance under overexposure conditions. Nonetheless, most methods still produce a satisfactory representation of the crack network.

In the quantification of soil crack geometric characteristics, DeepLabv3+ with ResNet-50 ranked highest in R_{SC} with the lowest error rate of 0.73 %, followed by Res-UNet (1.09 %) and DeepLabv3+ with ResNet-18 (3.33 %). U-Net had a significantly higher error rate of 32.03 % which was nearly 27 % higher than the second-lowest performer, DeepLabv3+ with MobileNetV2. All DeepLabv3+ models outperformed the other methods in w_{avg} with MobileNetV2 backbone leading at 0.8 % error. Errors were higher across all methods for w_{max}, with Otsu's and DeepLabv3+ with ResNet-50 showing the lowest errors (20.20 % and 26.91 % respectively). Moreover, DeepLabv3+ variants, particularly ResNet-18 and ResNet-50, excelled in L_{sum} computation with marginal error rates of 0.45 % and 0.58 % respectively. Despite having competitive performance in R_{SC}, Res-UNet lagged in L_{sum} and L_{avg} with error rates of 25.60 % and 92.78 %, respectively. DeepLabv3+ models performed better in L_{avg}, with error rates ranging between 31.17 % (ResNet-50) and 40.51 %

(Xception). For N_{int}, N_n, and N_{seg}, errors were significantly higher in traditional method, Res-UNet, and U-Net compared to DeepLabv3+ models, with DeepLabv3+ with MobileNetV2 showing the lowest errors in these parameters (57.14 % for N_{int}, 45.45 % for N_n, 38.89 % for N_{seg}), followed by ResNet-50 and ResNet-18 backbones.

The analysis of results from Case B underscores the effectiveness of advanced segmentation methods in addressing the challenges posed by overexposure. DeepLabv3+ with ResNet-50 emerged as the top performer with its high precision, recall, F1 score, and IoU values. It also demonstrated low errors in geometrical parameter estimation, indicating its robustness under the condition. All DeepLabv3+ variants, particularly those with MobileNetV2 and ResNets, maintained high accuracy in crack detection despite overexposure as shown by their high values across evaluation metrics and lower error rates in geometrical parameters quantification. While Res-UNet performed adequately in evaluation metrics, it exhibited higher errors in geometrical parameters computation compared to DeepLabv3+ models, highlighting its limitations in precise crack network reconstruction under such conditions. U-Net was notably less effective in crack segmentation, as evidenced by its high errors across all standards, reinforcing the need for more sophisticated architectures like DeepLabv3+ for accurate segmentation of soil cracks.

Table 4.7: Case B segmentation performance.

Method	Otsu's	U-Net	Res-UNet	DeepLabv3+ (MobileNetV2)	DeepLabv3+ (ResNet-18)	DeepLabv3+ (ResNet-50)	DeepLabv3+ (Xception)
Evaluation Metrics (%):							
Precision	92.30	65.87	91.02	93.30	93.51	92.91	93.75
Recall	87.67	86.97	90.02	88.51	90.40	93.60	89.33
F1 Score	89.93	74.96	90.52	90.84	91.93	93.25	91.49
IoU	81.70	59.95	82.68	83.22	85.06	87.36	84.31
Error on Geometrical Parameters (%):							
R_{SC}	5.02	32.03	1.09	5.13	3.33	0.73	4.71
w_{avg}	18.84	50.16	15.12	0.80	2.59	3.07	1.91
w_{max}	20.20	43.58	34.63	33.82	34.30	26.91	34.63
N_{int}	1185.71	40885.71	4242.86	57.14	85.71	85.71	114.29
N_n	500.00	13259.09	1381.82	45.45	54.55	45.45	72.73
N_{seg}	511.11	15844.44	1638.89	38.89	50.00	44.44	61.11
L_{sum}	6.89	236.85	25.60	5.59	0.45	0.58	4.16
L_{avg}	82.51	97.89	92.78	32.03	33.03	31.17	40.51

4.7.3 Case C: Blurry Image with Low Exposure

The image in Case C depicted a scenario characterised by slight blurriness accompanied by low exposure. The segmented masks are visualised in Figure 4.6. From these figures, it is evident that the segmentation masks produced by DeepLabv3+ models were more consistent with the ground truth, followed by Res-UNet. Otsu's method struggled to accurately generalise the crack lines due to its inability to incorporate spatial correlation information, which is crucial for handling blurry edges. In contrast, U-Net exhibited a significant reduction in noise compared to Case B. Coupled with the results from Case A, these observations suggest that U-Net performs better with low-exposure images, as indicated by the reduced noise issue in the figure.

The segmentation performance of various methods under Case C conditions is presented in Table 4.8. Otsu's method exhibited the highest precision at 98.27 %, while the deep learning methods achieved high precision scores ranging from 94.57 % (DeepLabv3+ with Xception) to 95.81 % (DeepLabv3+ with ResNet-18), with U-Net lagging significantly behind at 76.59 %. In terms of recall, DeepLabv3+ with Xception and ResNet-50 were the top performers, with only a 0.02 % difference between them. Otsu's method ranked lowest with a recall of 80.33 %. The high precision coupled with the relatively lower recall of Otsu's method suggests that it tends to produce incomplete segmentation, covering only a small, highly confident portion of the object while missing true positives. This is corroborated by the observation of fragmentary cracks in Otsu's mask shown in Figure 4.6.

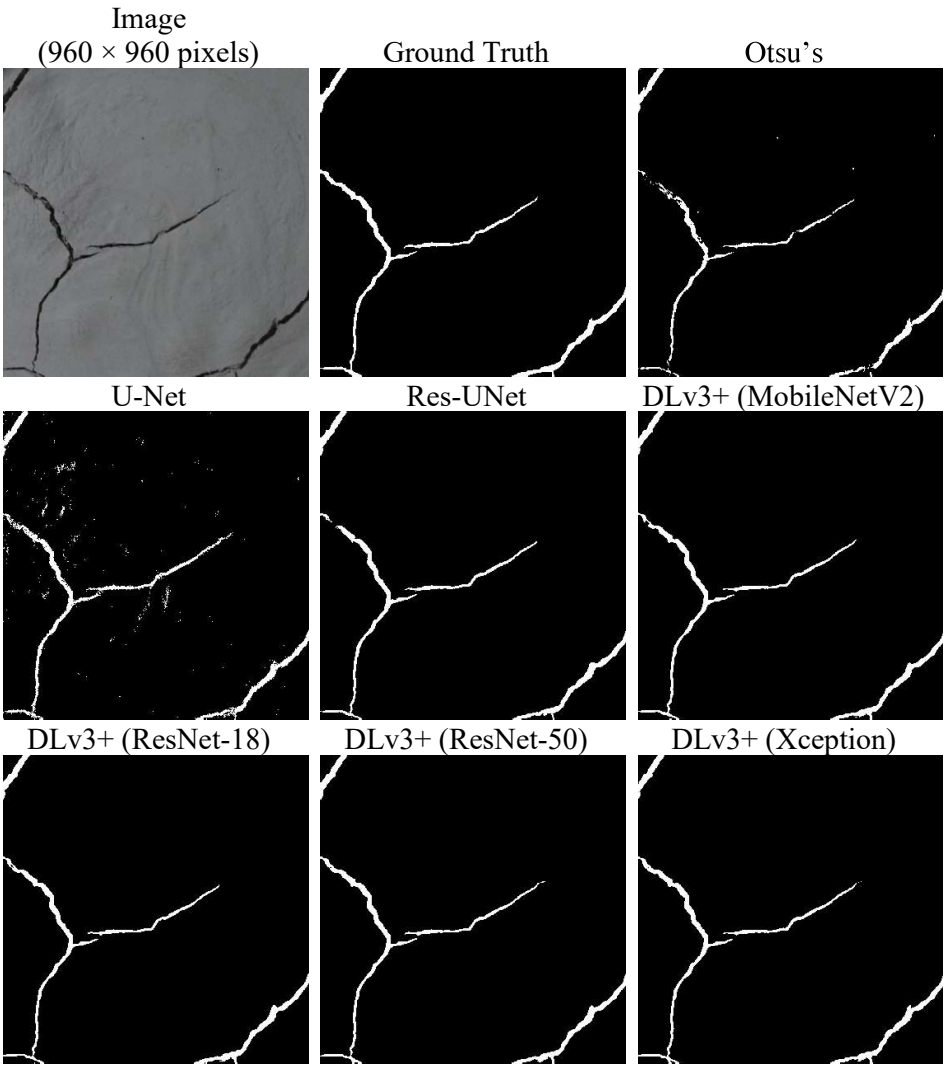


Figure 4.6: Segmentation visualisation for Case C.

For balanced performance, Res-UNet and DeepLabv3+ variants achieved relatively high F1 scores, with the DeepLabv3+ models featuring ResNet backbones leading with an F1 score of 95.41 %. DeepLabv3+ with ResNet-50 had the highest IoU at 91.23 %, followed by ResNet-18 with 91.22 % and the other DeepLabv3+ models. Despite a higher recall than Otsu's method, U-Net obtained a 10 % lower IoU (69.96 %) compared to Otsu's method (79.21 %) due to a higher incidence of false positives (noises).

In the analysis of geometrical parameters, DeepLabv3+ models excelled in R_{SC} and w_{avg} detection. ResNet-18 led in R_{SC} detection with a minimal error of 0.83 %, while Xception led in w_{avg} detection with an error of 0.59 %. DeepLabv3+ with ResNet-50 and Otsu's method both achieved zero error in w_{max} detection, followed by DeepLabv3+ with ResNet-18 and Xception. For crack intersections and segment detection, DeepLabv3+ models, particularly those with MobileNetV2, demonstrated significantly lower errors (50 % for N_{int}, 16.67 % for N_n, 25 % for N_{seg}) compared to traditional methods (750 % for N_{int}, 216.67 % for N_n, 237.50 % for N_{seg}), Res-UNet (3000 % for N_{int}, 483.33 % for N_n, 712.50 % for N_{seg}), and U-Net (45700 % for N_{int}, 7766.67 % for N_n, 11475 % for N_{seg}), indicating their superior ability to reconstruct crack networks under challenging conditions. All DeepLabv3+ models and Otsu's method exhibited marginal error rates in L_{sum}, with DeepLabv3+ with MobileNetV2 ranking first with an error of 0.16 % error. Although Otsu's method measured crack length with satisfactory accuracy, its performance in measuring average crack length was limited by its inability to compute N_{seg} accurately, resulting in a 70.93 % error in L_{avg}. DeepLabv3+ variants achieved moderate error rates in L_{avg}, with MobileNetV2 leading at 19.87 % and ResNet-18 closing at 41.78 %. Despite these moderate errors, DeepLabv3+ models still outperformed other methods due to their superior capability in detecting crack segments.

Combining visual observations with parameter interpretations, DeepLabv3+ architecture demonstrated its effectiveness in handling complex conditions, significantly outperforming other methods. It was able to

reconstruct the crack network with higher accuracy while preserving important geometric characteristics relevant to soil cracking studies as demonstrated by its lowest error in crack segment quantification. DeepLabv3+ with ResNet-50 and ResNet-18 emerged as the top performers in this case, achieving high scores in evaluation metrics and generally low errors across geometrical parameters. DeepLabv3+ with MobileNetV2 also performed well under limited computational resources, offering similar performance to ResNets-based models and presenting an excellent choice for practical applications. Res-UNet, while performing well in evaluation metrics, struggled in the accurate quantification of geometrical parameters (much higher error rates compared to DeepLabv3+), highlighting its limitations in precise crack network recognition. Otsu's method further exposed its limitations in handling blurry edges and low-exposure scenarios, emphasising the advantages of deep learning models that can interpret image context more broadly.

Table 4.8: Case C segmentation performance.

Method	Otsu's	U-Net	Res-UNet	DeepLabv3+ (MobileNetV2)	DeepLabv3+ (ResNet-18)	DeepLabv3+ (ResNet-50)	DeepLabv3+ (Xception)
Evaluation Metrics (%):							
Precision	98.27	76.59	95.14	95.80	95.81	94.91	94.57
Recall	80.33	89.00	93.02	94.63	95.01	95.92	95.94
F1 Score	88.40	82.33	94.07	95.21	95.41	95.41	95.25
IoU	79.21	69.96	88.81	90.86	91.22	91.23	90.92
Error on Geometrical Parameters (%):							
R_{SC}	18.25	16.21	2.22	1.21	0.83	1.06	1.45
w_{avg}	23.97	50.94	8.71	1.20	2.18	1.43	0.59
w_{max}	0.00	6.04	3.08	4.39	1.18	0.00	2.77
N_{int}	750.00	45700.00	3000.00	50.00	250.00	150.00	250.00
N_n	216.67	7766.67	483.33	16.67	50.00	33.33	50.00
N_{seg}	237.50	11475.00	712.50	25.00	75.00	50.00	75.00
L_{sum}	1.89	177.27	9.71	0.16	1.89	3.31	2.10
L_{avg}	70.93	97.60	86.50	19.87	41.78	31.13	41.66

4.7.4 Case D: Complex and Intersecting Crack Network

Case D presented a scenario involving a complex crack network characterised by numerous crack segments and intersections, as well as zigzag and fine cracks. Figure 4.7 illustrates the image and its corresponding binary masks generated by different methods. The traditional segmentation method effectively categorised pixels into their respective classes, though it suffered from some false negatives. This method has significant limitations, as it requires uniform illumination and a well-defined crack network with clear differentiation from background pixels. Figure 4.8 highlights these limitations by showing Otsu's method struggles with uneven illumination scenarios.

U-Net has also faced challenges with persistent noise, as evidenced in Figure 4.7. When comparing Res-UNet and DeepLabv3+ models, Res-UNet displayed a slight advantage in better matching the crack network to the ground truth. However, DeepLabv3+ models produced cleaner masks with minimal noise. Among the DeepLabv3+ variants, MobileNetV2 and ResNet-18 backbones demonstrated superior alignments with the ground truth and better crack connectivity, followed by ResNet-50 and Xception backbones.

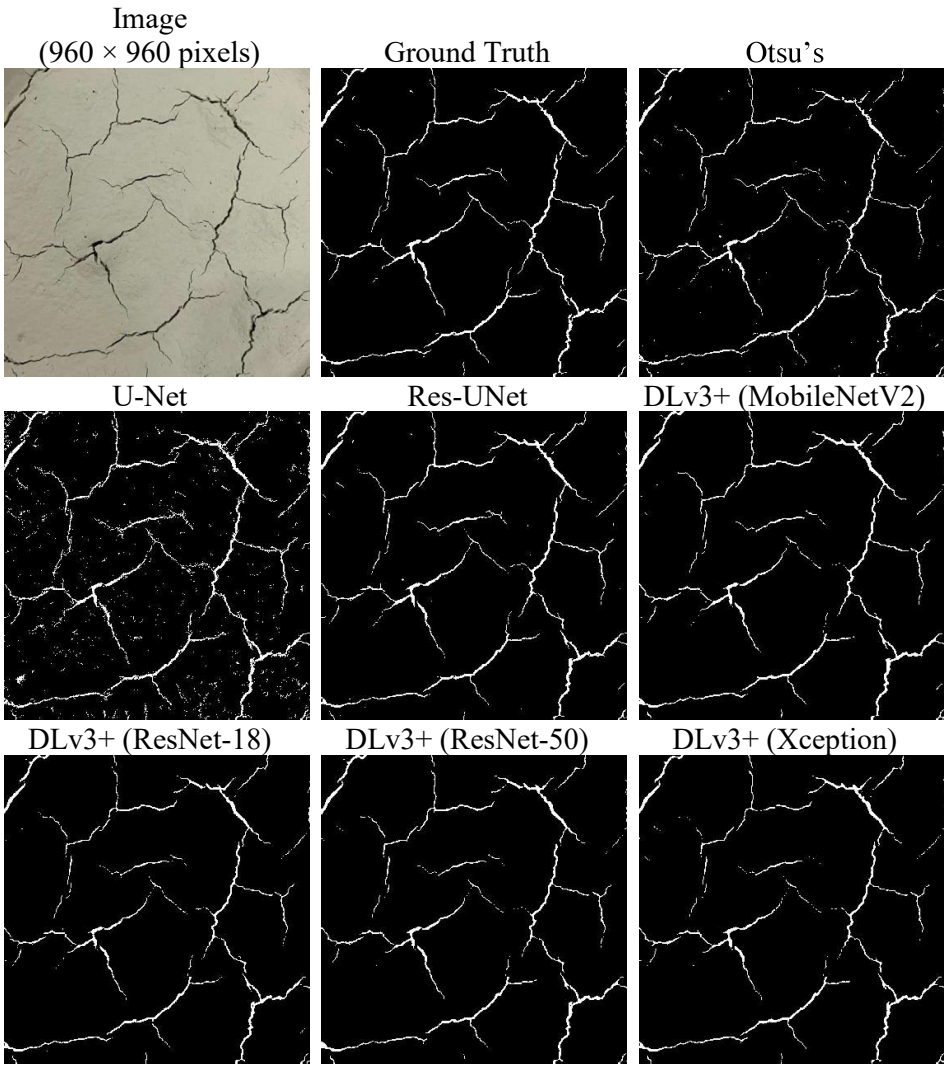


Figure 4.7: Segmentation visualisation for Case D.

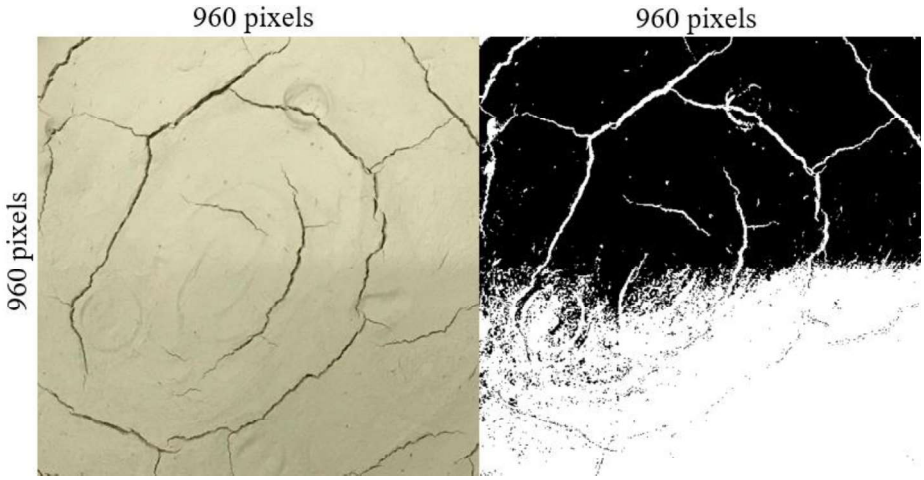


Figure 4.8: Otsu's mask on image with similar crack configuration as Case D but with uneven illumination.

Table 4.9 presents the segmentation performance by various methods for Case D, according to their respective standards. As indicated in the first part of the table, Otsu's method achieved the highest precision (91.96 %), while Res-UNet ranked first in recall (91.19 %), F1 score (89.59 %), and IoU (81.14 %). U-Net recorded the lowest values in precision (54.68 %), F1 score (66.86 %), and IoU (50.21 %), while DeepLabv3+ with Xception finished last with a recall of 82.62 %. Despite achieving a relatively high recall of 86.02 %, U-Net was categorised as the worst performer due to significant drawbacks in precision and IoU. The low precision and IoU values indicate U-Net's poor ability to produce a segmentation mask with satisfactory overlap with the ground truth, and its tendency to classify background pixels as cracks (high false positives). Although Otsu's method and Res-UNet led in performance, DeepLabv3+ variants demonstrated competitive performance across all evaluation metrics, with marginal differences from the leaders, showcasing their stability and reliability in handling various scenarios. Among them, DeepLabv3+ with

ResNet-18 achieved the highest precision (90.08 %), while DeepLabv3+ with MobileNetV2 achieved the highest recall (88.86 %), F1 score (87.59 %), and IoU (77.92 %).

Turning to the errors on geometrical parameters, as shown in the second part of the table, DeepLabv3+ with MobileNetV2 exhibited the lowest error in R_{SC} (2.91 %), closely followed by Res-UNet (3.59 %) and the remaining methods, except for U-Net, which had a significant error of 57.33 %. Regarding average crack width, Res-UNet performed exceptionally well with the lowest error of 0.67 %, while DeepLabv3+ variants ranged from 6.37 % (DeepLabv3+ with ResNet-18) to 13.99 % (DeepLabv3+ with MobileNetV2). Since the crack network displayed a relatively uniform crack width, all models performed well in w_{max} with some achieving zero error (Otsu's, Res-UNet, DeepLabv3+ with ResNet-18). Otsu's method recorded the lowest error rate in computing the number of intersections (38.71 %), closely followed by all DeepLabv3+ variants. For N_n and N_{seg}, DeepLabv3+ models achieved the lowest error rates, with ResNet-18 variant exhibiting zero error in N_n and 14.29 % error in N_{seg}. Despite the low errors in R_{SC} and crack width computations, Res-UNet struggled with accurate segmentation of individual crack segments and their intersections, with high error rates of 150 % in N_{seg} and 490.32 % in N_{int}. U-Net again faced challenges in crack segment recognition (error rates of 1981.63 %) due to persistent noise issues. Conversely, Res-UNet achieved the lowest error in L_{sum} (2.15 %), while DeepLabv3+ with Xception and ResNet-18 had the lowest errors in L_{avg} (around 3.10 %). The ability of Res-UNet to accurately measure

the average length of crack segments was hindered by its limitations in correctly identifying the number of segments, resulting in a higher error of L_{avg} (59.14 %).

The results for Case D demonstrate the ability of advanced segmentation methods to handle intricate crack networks effectively. DeepLabv3+ with MobileNetV2 and ResNet-18 showed exceptional performance in most evaluation metrics and geometrical parameters, providing better representations of the crack networks with higher accuracy in crack geometries. The robustness of DeepLabv3+ variants in recognising fine and zigzag cracks was evident from their relatively lower errors in parameters such as N_{int}, N_n, and N_{seg}. Conversely, Res-UNet exhibited strong recall and IoU values, indicating its proficiency in identifying large portions of actual cracks with accurate overlap. It also accurately quantified crack shapes and sizes, as indicated by the low errors in surface crack ratio and crack width computations. However, it was inadequate in recognising the connectivity of the crack segments, as evidenced by high errors in parameters such as N_{int}, N_n, and N_{seg}. U-Net revealed its limitations under the complexity of this case, with relatively low values across evaluation metrics and high errors in crack geometrical parameter measurements. Otsu's methods performed significantly better in geometrical parameters analysis with lower errors compared to cases B and C, but this performance was influenced by illumination constraints, as demonstrated in Figure 4.8.

Table 4.9: Case D segmentation performance.

Method	Otsu's	U-Net	Res-UNet	DeepLabv3+ (MobileNetV2)	DeepLabv3+ (ResNet-18)	DeepLabv3+ (ResNet-50)	DeepLabv3+ (Xception)
Evaluation Metrics (%):							
Precision	91.96	54.68	88.04	86.35	90.08	89.45	89.27
Recall	86.16	86.02	91.19	88.86	83.96	84.60	82.62
F1 Score	88.97	66.86	89.59	87.59	86.91	86.96	85.82
IoU	80.13	50.21	81.14	77.92	76.85	76.93	75.16
Error on Geometrical Parameters (%):							
R_{SC}	6.31	57.33	3.59	2.91	6.79	5.42	7.44
w_{avg}	5.06	25.68	0.67	13.99	6.37	12.36	11.53
w_{max}	0.00	5.00	0.00	4.76	0.00	4.76	4.76
N_{int}	38.71	6416.13	490.32	45.16	48.39	45.16	41.94
N_n	56.52	1470.29	110.14	17.39	0.00	15.94	15.94
N_{seg}	39.80	1981.63	150.00	25.51	14.29	25.51	23.47
L_{sum}	10.33	76.66	2.15	12.46	16.98	19.45	21.10
L_{avg}	35.86	91.51	59.14	17.51	3.15	8.14	3.10

4.7.5 Case E: Low Exposure with Ill-defined Crack Edges

The scenario in Case E was characterised by a crack network with ill-defined crack edges and lines, along with low exposure conditions. The masks segmented through various methods are visualised in Figure 4.9. It is evident from the figure that traditional methods (such as Otsu's) and U-Net were inadequate in handling crack networks with indistinct edges. U-Net consistently struggled with noise issues, while Otsu's method failed to accurately segment the crack edges, resulting in a crack network with ambiguous shapes. More sophisticated models, including Res-UNet and DeepLabv3+ variants, provided segmentations that more closely resembled the ground truth. Among these, DeepLabv3+ variants demonstrated superior performance by producing segmentation with fewer holes (black pixels enclosed by crack pixels). This improvement is attributed to their ability to identify pixel relationships through enlarged receptive fields, which helped to preserve spatial information.

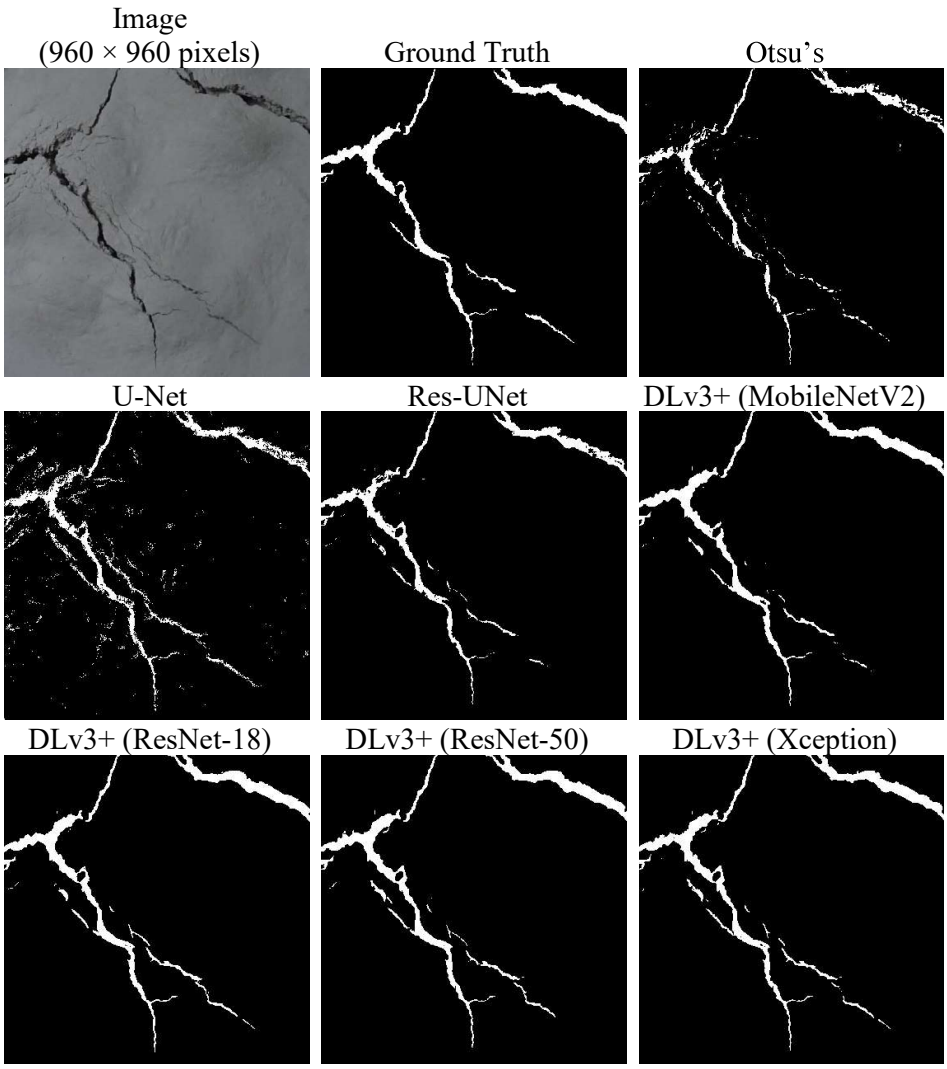


Figure 4.9: Segmentation visualisation for Case E.

Table 4.10 presents the segmentation performance for Case E. In terms of standard metrics, Otsu's method achieved the highest precision of 93.17 %, followed by Res-UNet at 90.36 %, and DeepLabv3+ models ranging from 82.35 % (ResNet-50) to 86.38 % (MobileNetV2). These methods effectively identified crack pixels under conditions of poor edge definition, as indicated by their high precision values and minimal false positives. DeepLabv3+ models, particularly with ResNet-18 and ResNet-50, achieved the highest recall values of 96.28 % and 95.87 %, respectively. In contrast, Otsu's method and Res-UNet,

which excelled in precision, had lower recall values of 70.37 % and 83.38 %, respectively. This discrepancy highlights that DeepLabv3+ models were more effective in capturing most of the actual cracks present in the image, while Otsu's method and Res-UNet missed more crack pixels, resulting in a combination of high precision and low recall. Consequently, Otsu's method and Res-UNet had slightly lower F1 scores compared to DeepLabv3+ models. DeepLabv3+ models achieved high F1 scores, ranging from 88.60 % (ResNet-50) to 89.68 % (Xception), effectively balancing precision and recall. DeepLabv3+ with Xception achieved the highest IoU (81.29 %), followed closely by other Deeplabv3+ variants. This signifies the superior performance of DeepLabv3+ models in generating segmentation outputs that closely match the ground truth.

In the second part of Table 4.10, Res-UNet and DeepLabv3+ with MobileNetV2 exhibited the lowest errors in R_{SC} at 7.72 % and 7.84 %, respectively. Otsu's method had the highest error in R_{SC} (24.47 %), due to a higher number of false negative predictions. DeepLabv3+ models showed significantly lower error rates in both average and maximum crack width measurements compared to other methods. DeepLabv3+ with Xception had the lowest error in w_{avg}, at a marginal value of 0.56 %, followed by ResNet-18, ResNet-50, and MobileNetV2 variants. These models also achieved minimal errors in $w_{max,}$ ranging from 4.52 % (Xception backbone) and 9.22 % (MobileNetV2 backbone). The number of intersections and nodes can significantly affect the measurement of crack segments. In this case, DeepLabv3+ demonstrated superior capability in segmenting the crack network

with accurate connectivity between crack lines. It achieved substantially lower error rates across N_{int}, N_n, and N_{seg} compared to other methods. The lowest error in N_{int} was recorded by ResNet-50 variant at 50 %, while U-Net had an exceptionally high error of 57525 %. DeepLabv3+ with ResNet-50 and MobileNetV2 showed the lowest error of 10 % in N_n parameter. Combined with their lower errors in N_{int}, they achieved the lowest error in N_{seg} at 7.14 % and 14.29 %, respectively. Among the DeepLabv3+ models, ResNet-18 and Xception backbones exhibited higher errors in N_{int} and N_n, resulting in higher errors in the final computation of N_{seg}. DeepLabv3+ with ResNet-50 ranked first in both L_{sum} and L_{avg}, with marginal errors of 2.46 % and 4.37 % respectively, significantly outperforming Otsu's method, U-Net, and Res-UNet.

The results from Case E highlight the strengths and weaknesses of each method under the challenging condition of ill-defined crack edges. U-Net consistently faced issues with noise, leading to low accuracy in terms of evaluation metrics and high errors in geometrical parameter measurements. Although Otsu's method performed well in terms of precision and L_{sum}, it was inadequate under this scenario due to its low IoU and higher errors in other geometrical parameters. Res-UNet demonstrated strong performance in evaluation metrics with comparatively high IoU and precision, and geometrical parameters in terms of R_{SC} but struggled with elevated errors in N_{int} and N_{seg}, which are crucial for describing crack characteristics. DeepLabv3+ with Xception stood out in evaluation metrics analysis with the highest scores in F1 score and IoU. It achieved the lowest errors in w_{avg} and w_{max} measurements but fell short in N_{int}, N_n, N_{seg}, L_{sum}, and L_{avg} compared to ResNet-50 variant. In

summary, DeepLabv3+ with ResNet-50 and Xception emerged as the most reliable methods for crack segmentation in images with ill-defined crack edges and low exposure.

Table 4.10: Case E segmentation performance.

Method	Otsu's	U-Net	Res-UNet	DeepLabv3+ (MobileNetV2)	DeepLabv3+ (ResNet-18)	DeepLabv3+ (ResNet-50)	DeepLabv3+ (Xception)
Evaluation Metrics (%):							
Precision	93.17	64.10	90.36	86.38	83.25	82.35	84.88
Recall	70.37	78.22	83.38	93.15	96.28	95.87	95.06
F1 Score	80.18	70.46	86.73	89.64	89.29	88.60	89.68
IoU	66.92	54.39	76.57	81.22	80.66	79.53	81.29
Error on Geometrical Parameters (%):							
R_{SC}	24.47	22.02	7.72	7.84	15.65	16.42	11.99
w_{avg}	40.32	68.00	34.99	11.16	5.68	7.16	0.56
w_{max}	25.14	38.74	22.61	9.22	6.59	5.50	4.52
N_{int}	1300.00	57525.00	10300.00	125.00	350.00	50.00	450.00
N_n	300.00	11905.00	2100.00	10.00	80.00	10.00	80.00
N_{seg}	378.57	16450.00	2942.86	14.29	100.00	7.14	107.14
L_{sum}	5.72	400.56	94.47	7.90	12.83	2.46	17.19
L_{avg}	77.91	96.98	93.61	19.41	43.59	4.37	43.42

4.7.6 Case F: Shadowed Soil Surface (Uneven Illumination)

Case F involved an image with uneven illumination on a smooth soil surface, including distinct shadows cast over one side of the image. Figure 4.10 shows the original image, its ground truth, and segmentation masks produced by various methods. From the figure, it is evident that Otsu's method and U-Net encountered difficulties in handling images with varying illumination conditions. Both methods struggled to differentiate between shadows and cracks, as these regions often exhibited similarly low-intensity values. Otsu's method, due to its threshold selection process, erroneously classified nearly the entire shadowed region as crack pixels. The shadowy areas, which had intensity values similar to the soil cracks, were inaccurately identified as cracks.

U-Net, while an improvement over traditional methods, still had limitations in distinguishing shadow edges from cracks due to its relatively simple feature extraction process. Despite significant advancements, U-Net occasionally misclassified the edges of shadow regions as crack pixels. In contrast, Res-UNet and DeepLabv3+ models performed excellently under varying illumination conditions, as demonstrated by their accurate crack segmentations in Figure 4.10. These models exhibited superior capabilities in identifying crack networks precisely, even in the shadowed regions of the image.

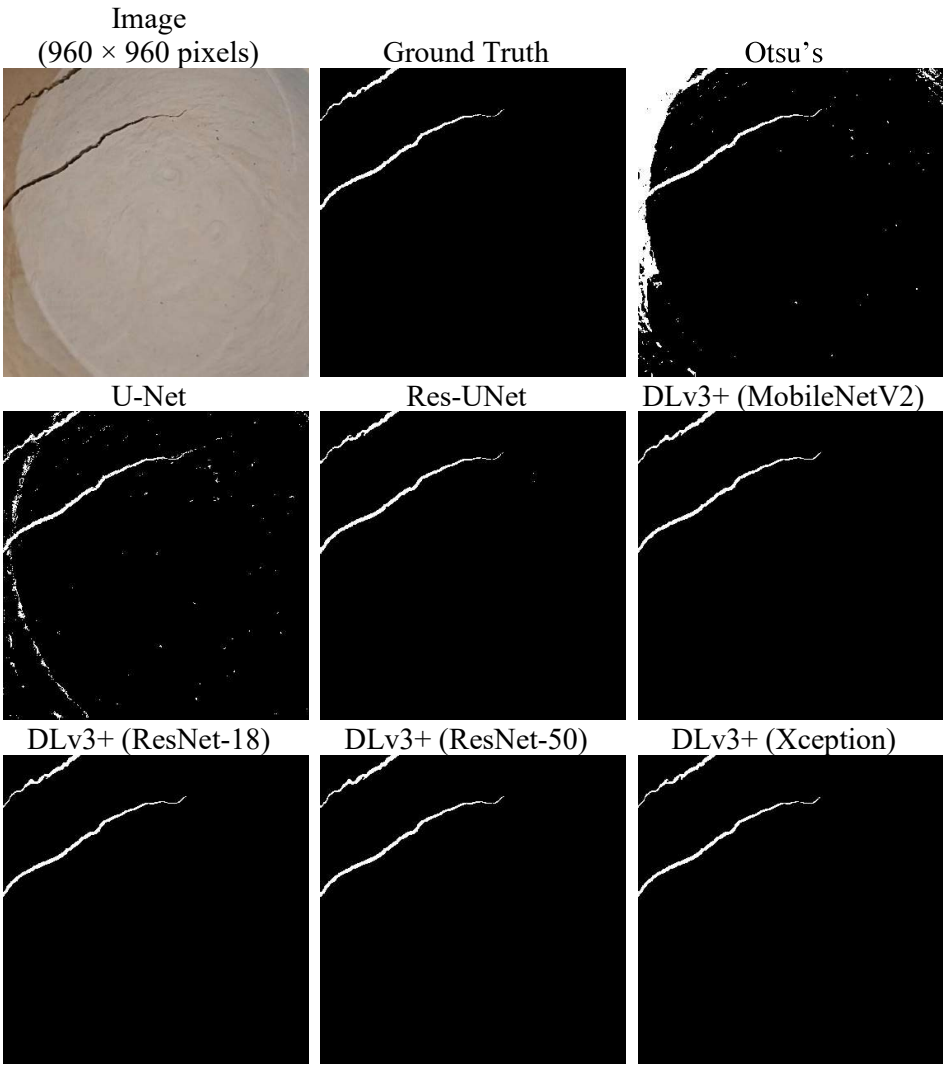


Figure 4.10: Segmentation visualisation for Case F.

The quantitative analysis of segmentation performance by various methods is shown in Table 4.11. For evaluation metrics, Res-UNet and DeepLabv3+ variants performed exceptionally well in precision, all scoring over 93 % and DeepLabv3+ with ResNet-18 leading at 95.85 %. The significantly reduced precision by Otsu's method (15.43 %) correlates with the visual observation described earlier, where most shadowed regions were incorrectly classified as cracks, resulting in high false positives. All methods achieved high recall values, well over 93 %, with Otsu's method ranking first

at 99.04 %, followed by DeepLabv3+ with ResNet-50 at 98.21 %. It is important to highlight that the high recall in Otsu's method does not necessarily reflect its effectiveness in capturing actual crack instances but rather results from classifying most shadow regions as crack pixels. Since some cracks overlapped with the shadow regions, this tendency greatly reduced the number of false negatives, resulting in a high recall for Otsu's method. DeepLabv3+ with ResNet-18 achieved the highest F1 score (96.08 %), closely followed by other DeepLabv3+ variants and Res-UNet, indicating the effectiveness of deep learning methods in balancing precision and recall. Res-UNet and DeepLabv3+ models performed similarly in IoU, with DeepLabv3+ with ResNet-18 leading at 92.45 %, while other models scored above 91 %. These high scores reflect the models' superior matching with the ground truth crack networks.

In the geometrical parameters analysis, DeepLabv3+ with ResNet-18 and Xception achieved a minimal error of 0.48 % and 0.82 % respectively for R_{SC}, suggesting accurate quantification of crack shapes and sizes. For average crack width, DeepLabv3+ with Xception recorded the lowest error of 1.72 %, followed closely by Res-UNet and DeepLabv3+ with ResNet-18. All deep learning models performed well with marginal errors in w_{max}, with Res-UNet and DeepLabv3+ with ResNet-18 achieving zero error rates. Otsu's method had the highest error across these parameters (541.80 % in R_{SC}, 119.57 % in w_{avg}, 530.41 % in w_{max}), largely due to its misclassification of shadow regions. Regarding crack line connectivity, DeepLabv3+ models outperformed others, particularly MobileNetV2 and Xception backbones, which achieved zero errors in N_{int}, N_n, and N_{seg}. Res-UNet retained a competitive edge in L_{sum} with a low

error of 1.26% but fell behind in L_{avg} (81.59 %) due to higher errors in crack segment measurements. Among the DeepLabv3+ variants, both ResNet backbones achieved lower error rates in L_{sum} (0.11 % for ResNet-18 and 0.23 % for ResNet-50) compared to their counterparts, while MobileNetV2 and Xception backbones excelled in L_{avg} (2.29 % and 1.72 % respectively) due to their higher accuracy in N_{seg}.

The ability to handle uneven illumination conditions is crucial for segmentation methods due to the ubiquitous presence of shadows. The results show that Otsu's method and U-Net were not well-suited for such challenging conditions, as evidenced by their inaccurate segmented masks and higher errors across most metrics. Traditional segmentation methods using global thresholding algorithms like Otsu's demonstrated a lack of flexibility in complex image conditions. More sophisticated architecture such as Res-UNet, significantly improves accuracy with their capability to generate masks that more closely align with the ground truth. Res-UNet also performed strongly in geometrical parameters, including R_{SC}, w_{avg}, w_{max}, and L_{sum}, showcasing its proficiency in identifying crack shapes and sizes. However, it struggled with connectivity issues, as indicated by higher errors in N_{int}, N_n, and N_{seg}. DeepLabv3+ models excelled in both evaluation metrics and geometrical parameters, emerging as the most reliable methods for soil crack segmentation under uneven illumination. Among these, ResNet-18 and ResNet-50 were slightly better choices if IoU and precision were primary concerns. However, MobileNetV2 and Xception backbones also performed well, showing minimal

errors in crack intersections and segment identification, even in the presence of shadows.

Table 4.11: Case F segmentation performance.

Method	Otsu's	U-Net	Res-UNet	DeepLabv3+ (MobileNetV2)	DeepLabv3+ (ResNet-18)	DeepLabv3+ (ResNet-50)	DeepLabv3+ (Xception)
Evaluation Metrics (%):							
Precision	15.43	49.13	93.73	93.27	95.85	93.12	95.82
Recall	99.04	93.72	97.18	97.44	96.31	98.21	95.03
F1 Score	26.70	64.46	95.42	95.31	96.08	95.60	95.42
IoU	15.41	47.56	91.24	91.04	92.45	91.57	91.25
Error on Geometrical Parameters (%):							
R_{SC}	541.80	90.77	3.68	4.47	0.48	5.46	0.82
w_{avg}	119.57	48.31	1.73	7.75	2.71	7.01	1.72
w_{max}	530.41	4.41	0.00	5.05	0.00	2.55	7.15
N_{int}	15900.00	63400.00	800.00	0.00	200.00	200.00	0.00
N_n	4150.00	16575.00	250.00	0.00	100.00	50.00	0.00
N_{seg}	8050.00	30650.00	450.00	0.00	150.00	100.00	0.00
L_{sum}	292.44	285.22	1.26	2.29	0.11	0.23	1.72
L_{avg}	95.18	98.75	81.59	2.29	60.05	49.89	1.72

4.7.7 Case G: Uneven Illumination with Fissure Cracks and Surface Aggregation

Case G involved an image with uneven illumination, featuring surface fissure cracks and small soil aggregates around the crack lines on the soil surface. Figure 4.11 illustrates the image and the corresponding segmented masks for the case.

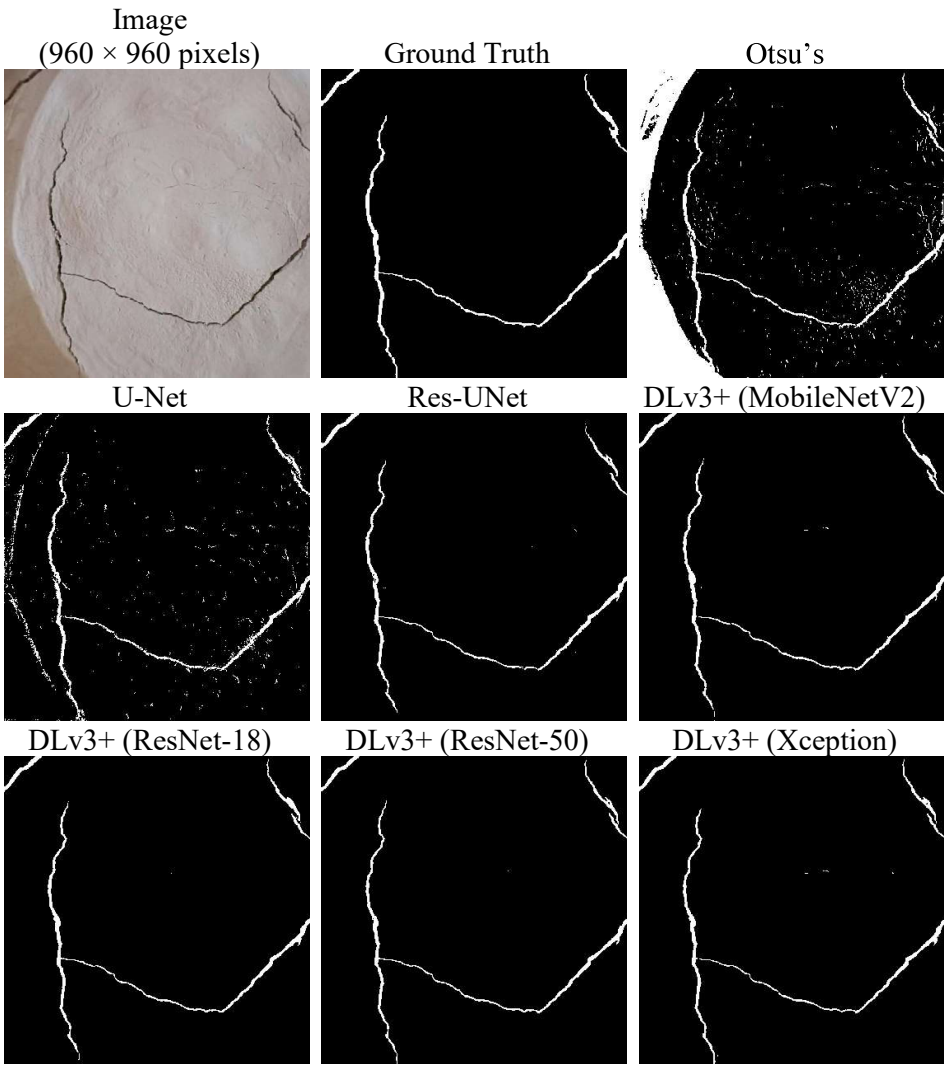


Figure 4.11: Segmentation visualisation for Case G.

Regarding uneven illumination, Otsu's method once again struggled to differentiate shadow from actual cracks due to the shadows' higher intensity values. Additionally, Otsu's method misclassified small aggregates and fissure cracks on the soil surface, which also had darker colours, as crack pixels. U-Net performed similarly to its performance in Case F, where it segmented only the shadow's edge as the foreground while exhibiting persistent noise problems. In contrast, Res-UNet and DeepLabv3+ variants significantly outperformed the previous methods, providing a closer representation of the actual crack network. A closer inspection reveals that DeepLabv3+ had a slight edge over Res-UNet, with better connectivity in crack lines and more precise crack edge definitions.

Table 4.12 presents the segmentation performance for Case G across various evaluation standards. From evaluation metrics analysis, DeepLabv3+ with ResNet-50 achieved the highest precision of 97.76 %, indicating its ability to correctly identify crack pixels with minimal false positives. Other DeepLabv3+ variants and Res-UNet also performed excellently, with precision values well over 96 %. Otsu's method displayed similar behaviour to Case F, characterised by low precision and high recall, due to a high rate of falsely classified background pixels that reduced false negatives. Among the remaining methods, DeepLabv3+ with ResNet-18 achieved the highest recall (88.68 %), followed closely by DeepLabv3+ with MobileNetV2 at 88.11 %. This indicates the effectiveness of these DeepLabv3+ models in capturing most of the actual cracks in the image. All DeepLabv3+ models obtained high F1 scores, with the ResNet-18 backbone leading at a value of 92.52 %, demonstrating a strong balance between precision and recall. In terms of crack network overlap,

DeepLabv3+ models achieved top performances, with ResNet-18 variant leading with an IoU of 86.09 %, indicating its satisfactory alignment with the ground truth crack segments.

In the quantifications of geometrical parameters, DeepLabv3+ with MobileNetV2 and ResNet-18 showed the lowest errors of 8.22 % and 8.31 % respectively in R_{SC}, showcasing their ability to quantify crack intensity with minimal errors in complex scenarios. Additionally, DeepLabv3+ with ResNet-18 achieved the lowest errors in both average and maximum crack width measurements at 8.73 % and 5.66 % respectively, with MobileNetV2 variant coming second with slightly higher error rates at 10.69 % and 7.70 % respectively. Otsu's method again demonstrated excessive error rates across these parameters (421.45 % for R_{SC}, 129.98 % for w_{avg}, 705.29 % for w_{max}) due to its inability to differentiate shadowy regions from actual cracks. U-Net performed the worst among deep learning methods in these parameters due to its noise issue, with error rates of 46.49 %, 53.80 %, and 20 % for R_{SC}, w_{avg}, and w_{max} respectively. Although Res-UNet performed well, it generally showed higher error rates compared to DeepLabv3+ variants.

For N_{int}, N_n, and N_{seg}, which represent the connectivity of crack lines, DeepLabv3+ with MobileNetV2 had zero error in N_{int} and significantly lower error rates for N_n (50 %) and N_{seg} (40 %) than the other methods, demonstrating robustness in detecting intersections and individual crack segments despite complex surface features. The other DeepLabv3+ variants also demonstrated superior performance with substantially lower errors across these parameters

compared to Res-UNet, Otsu's method, and U-Net, where U-Net see the highest errors of 109700 % in N_{int}, 14487.50 % in N_n, and 21680 % in N_{seg}. In crack length analysis, DeepLabv3+ with ResNet-18 excelled in L_{sum} with zero error, followed closely by DeepLabv3+ with MobileNetV2 with a marginal error of 0.48 %. Due to its better performance in N_{seg}, DeepLabv3+ with MobileNetV2 outperformed ResNet-18 variant in L_{avg} analysis at 28.92 % despite having slightly higher error in L_{sum}.

Compared to Case F, Case G presented a more challenging scenario with the presence of soil clods and surface fissures in addition to uneven illumination. The results indicated that Otsu's method was consistently troubled by uneven illumination and U-Net by noise issues, making both unsuitable for segmentation tasks under challenging conditions. Res-UNet showed significantly stronger performance in both evaluation metrics and geometrical parameters analysis compared to the previous two methods, reflecting its higher accuracy in quantifying crack shapes and sizes. However, it struggled with parameters related to crack network connectivity (N_{int}, N_n, and N_{seg}), undermining its reliability in soil crack segmentation. DeepLabv3+ variants demonstrated the strongest performances across all evaluation criteria. Considering both evaluation metrics and crack geometrical characteristics, DeepLabv3+ with ResNet-18 and MobileNetV2 emerged as the most reliable methods for segmenting soil cracks in images with uneven illumination and surface fissures. Their high performance across these standards suggests their robustness and applicability in challenging conditions.

Table 4.12: Case G segmentation performance.

Method	Otsu's	U-Net	Res-UNet	DeepLabv3+ (MobileNetV2)	DeepLabv3+ (ResNet-18)	DeepLabv3+ (ResNet-50)	DeepLabv3+ (Xception)
Evaluation Metrics (%):							
Precision	17.10	57.69	96.92	96.01	96.71	97.76	96.82
Recall	89.15	84.51	84.69	88.11	88.68	86.23	86.44
F1 Score	28.69	68.57	90.39	91.89	92.52	91.63	91.33
IoU	16.75	52.17	82.47	85.00	86.09	84.56	84.05
Error on Geometrical Parameters (%):							
R_{SC}	421.45	46.49	12.62	8.22	8.31	11.79	10.72
w_{avg}	129.98	53.80	19.26	10.69	8.73	14.32	13.00
w_{max}	705.29	20.00	17.54	7.70	5.66	20.00	10.56
N_{int}	15100.00	109700.00	7700.00	0.00	100.00	700.00	900.00
N_n	1950.00	14487.50	950.00	50.00	75.00	125.00	150.00
N_{seg}	3020.00	21680.00	1500.00	40.00	80.00	180.00	220.00
L_{sum}	128.98	199.86	10.70	0.48	0.00	5.35	4.44
L_{avg}	92.66	98.62	93.08	28.92	44.44	62.37	67.36

4.7.8 Case H: Defective Soil Surface with Shadows and Holes

Case H involved an image with shadows and small holes on the soil surface caused by bubbly defects. Figure 4.12 illustrates the image and the masks generated by various segmentation methods. The figure clearly shows that traditional segmentation methods, such as Otsu's algorithm, are inferior in handling unevenly illuminated images. Otsu's method classified almost the entire shadowy region as foreground. In contrast, deep learning methods, even the simplest U-Net network, demonstrated exceptional capability in handling images with uneven illumination, as evidenced by their segmented masks. Although U-Net was consistently plagued with noise speckles, it did not mistake the shadow's edge as crack lines, unlike in Cases F and G. This is likely due to U-Net's focus on capturing edge-like features, which helped it to differentiate between shadows and actual cracks when the shadow did not have a clearly defined border. Both Res-UNet and DeepLabv3+ variants produced masks with a high resemblance to the ground truth, indicating their robustness in generalising complex soil crack networks. A closer examination of the figure reveals that DeepLabv3+ models provided better representations of the ground truth crack network, while the Res-UNet mask contained more minor noise speckles.

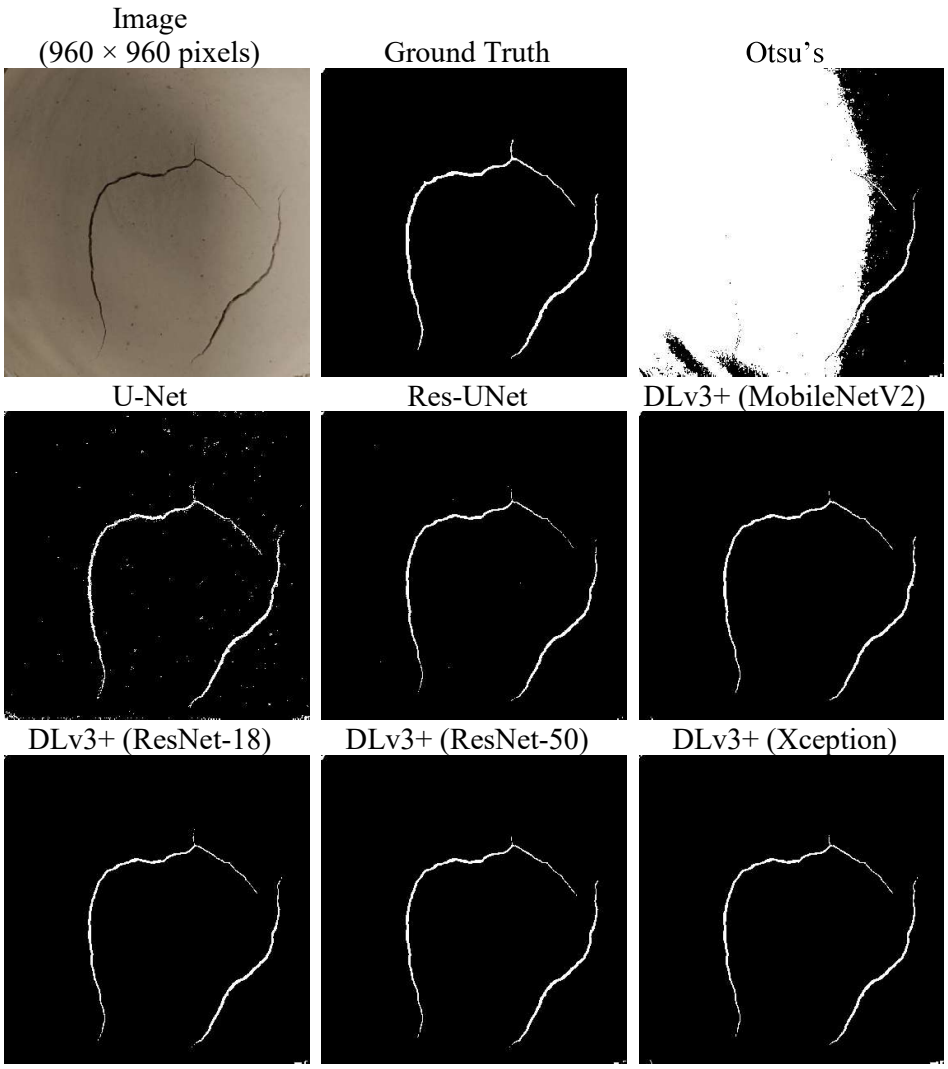


Figure 4.12: Segmentation visualisation for Case H.

The quantitative analysis of the segmentation performance across models is summarised in Table 4.13. In terms of evaluation metrics, Res-UNet achieved a superior precision of 99.06 %, followed closely by all the DeepLabv3+ variants. Otsu's method obtained the highest recall; however, this was due to a high number of false positive instances, as previously noted. Among the deep learning methods, DeepLabv3+ with ResNet-50 had the highest recall at 82.84 %, while Res-UNet had the lowest recall at 75.91 %. This indicates the effectiveness of DeepLabv3+ models, particularly with ResNet-50

backbone, in capturing the majority of the actual cracks despite the presence of shadows and holes. DeepLabv3+ with ResNet-50 also achieved the highest F1 score of 89.66 %, balancing precision and recall effectively. Furthermore, the ResNet-50 variant achieved the highest IoU at 81.25 %, followed by MobileNetV2 variant at 78.86 %, demonstrating better alignment with ground truth compared to other models. The generally lower IoU range in this case, compared to previous cases, is primarily due to the fewer crack pixels in the image, which exaggerated the errors.

In terms of geometrical parameters, U-Net showed the lowest error (14.93 %) in R_{SC}, followed by DeepLabv3+ with ResNet-50 (15.21 %). The other DeepLabv3+ variants also performed well, with error rates around 18 %. Res-UNet had the highest R_{SC} error at 23.37 % among the deep learning methods. For average crack width, DeepLabv3+ with ResNet-50 obtained the lowest error at 15.56 %, while other DeepLabv3+ variants performed similarly, with a $2-3$ % difference in error rates. DeepLabv3+ models outperformed others in w_{max} measurement with consistent errors of 16.70 %. In terms of crack line connectivity, DeepLabv3+ models excelled in crack intersection detection with zero error. For N_n and N_{seg}, DeepLabv3+ with Xception had the lowest error rates at 33.33 % and 25 % respectively, followed by other variants. Res-UNet and DeepLabv3+ models demonstrated superior capability in total crack length measurements with minimal error rates, led by DeepLabv3+ with ResNet-18 (0.31 %) and MobileNetV2 (0.56 %). For similar error rates, DeepLabv3+ with Xception ranked highest in L_{avg} computation due to its superior accuracy in individual crack segment segmentation.

The results underscore the limitations of traditional segmentation methods in handling images with varying conditions, which are common in real-life scenarios. Advanced deep learning methods demonstrated significant improvements in segmenting images with shadows. However, simple architectures like U-Net still struggled to generate accurate representations of crack networks as demonstrated by its lower IoU value. Res-UNet showed strong performance in precision but only obtained moderate results in other metrics and geometrical parameters. DeepLabv3+ with ResNet-50 excelled in evaluation metrics, indicating its robust ability to accurately identify and capture crack pixels under challenging conditions. It also performed well in quantifying surface crack ratio and crack width but struggled slightly with individual crack segment detection. Overall, DeepLabv3+ with ResNet-50 provided the best segmentation performance in this case, with high accuracy across metrics like IoU and precision. For scenarios where computational resources are a concern, DeepLabv3+ with MobileNetV2 offers an excellent segmentation solution, providing comparable performance at a significantly lower computational cost.

Table 4.13: Case H segmentation performance.

Method	Otsu's	U-Net	Res-UNet	DeepLabv3+ (MobileNetV2)	DeepLabv3+ (ResNet-18)	DeepLabv3+ (ResNet-50)	DeepLabv3+ (Xception)
Evaluation Metrics (%):							
Precision	2.00	69.73	99.06	98.42	97.52	97.70	97.20
Recall	95.23	80.15	75.91	79.87	79.28	82.84	79.36
F1 Score	3.92	74.58	85.95	88.18	87.46	89.66	87.38
IoU	2.00	59.46	75.37	78.86	77.71	81.25	77.58
Error on Geometrical Parameters (%):							
R_{SC}	4661.18	14.93	23.37	18.85	18.71	15.21	18.35
w_{avg}	854.88	47.40	21.31	17.49	18.79	15.56	18.14
w_{max}	2733.51	19.19	23.07	16.70	16.70	16.70	16.70
N_{int}	109100.00	54100.00	1200.00	0.00	0.00	0.00	0.00
N_n	18366.67	9283.33	200.00	66.67	66.67	100.00	33.33
N_{seg}	27575.00	13300.00	300.00	50.00	50.00	75.00	25.00
L_{sum}	1370.16	103.75	2.18	0.56	0.31	1.12	1.00
L_{avg}	94.69	98.48	75.55	33.71	33.13	42.22	20.80

4.7.9 Case-wise Segmentation Performance Summary

DeepLabv3+ with MobileNetV2 was suggested as the best segmentation model according to accuracy and stability indicators in section 4.6. Therefore, a summary of its performance across evaluation metrics and error rates on geometrical parameters quantification on various case studies is presented in **Table 4.14**, and its accuracy and stability indexes across various evaluation standards are presented in Table 4.15.

The model achieved its best performance in Case A, which involved a standard crack network with a clear boundary definition. This case recorded the highest F1 Score of 95.66 % and IoU of 91.69 %, along with the highest mean across evaluation metrics (94.67 %) and the lowest mean in error rates for geometrical parameters quantification analysis (1.93 %). It showcased the model's ability to perform optimally under ideal imaging conditions.

In contrast, performance declined in Cases D, E, and H, which involved images with complex intersecting crack networks, low exposure with ill-defined crack edges, and defective soil surface with shadows and holes respectively. These cases showed about a 10 % drop in IoU values (77.92 %, 81.22 %, and 78.86 %) compared to Case A. The mean error rates across selected geometrical parameters for these cases also increased to 14.91 %, 12.60 %, and 25.90 % respectively, suggesting that the model struggles with intricate geometries and poor contrast.

The model resulted in acceptable segmentation performance in Case B (overexposure) but exhibited exaggerated crack width and crack connectivity predictions, as shown by their error rates of 33.82 % for w_{max} and 38.89 % for N_{seg} respectively. This might be pointing to the model's sensitivity to excessive illumination which affects crack boundaries recognition. In contrast, Case C, which features low exposure, produced reliable results, with a high mean score of 94.13 % across evaluation metrics and a low mean score of 7.99 % across geometrical parameters quantification error. This suggested that the model could generalise well under low exposure conditions.

On the other hand, regarding Case F with shadowed and unevenly illuminated surface, the model maintained relatively strong F1 Score (95.31 %) and IoU (91.04 %) values, suggesting that the model can handle images with the presence of shadows. This robustness is also demonstrated in Case G, albeit the performance was slightly hindered due to surface aggregation and fissure-like textures. In terms of geometrical parameters quantification, the model confirmed its ability in handling shadowy conditions with relatively low error rates across parameters, with Case F achieving 4.89 % mean error rates.

Overall, while DeepLabv3+ with MobileNetV2 demonstrated high segmentation accuracy under standard and moderately distorted conditions, performance declined when faced with complex crack geometries, surface anomalies, and excessive illumination.

Table 4.14: Summary of case-wise segmentation performance of best performing model.

DeepLabv3+ (MobileNetV2)	CASE							
	A	B	C	D	E	F	G	H
Evaluation Metrics (%):								
Precision	95.55	93.30	95.80	86.35	86.38	93.27	96.01	98.42
Recall	95.77	88.51	94.63	88.86	93.15	97.44	88.11	79.87
F1 Score	95.66	90.84	95.21	87.59	89.64	95.31	91.89	88.18
IoU	91.69	83.22	90.86	77.92	81.22	91.04	85.00	78.86
Error on Geometrical Parameters (%):								
R_{SC}	0.23	5.13	1.21	2.91	7.84	4.47	8.22	18.85
w_{avg}	2.78	0.8	1.20	13.99	11.16	7.75	10.69	17.49
w_{max}	2.23	33.82	4.39	4.76	9.22	5.05	7.70	16.70
N_{int}	0.00	57.14	50.00	45.16	125.00	0.00	0.00	0.00
N_n	0.00	45.45	16.67	17.39	10.00	0.00	50.00	66.67
N_{seg}	0.00	38.89	25.00	25.51	14.29	0.00	40.00	50.00
L_{sum}	2.46	5.59	0.16	12.46	7.90	2.29	0.48	0.56
L_{avg}	2.46	32.03	19.87	17.51	19.41	2.29	28.92	33.71

Table 4.15: Accuracy and stability of the best performing model across various standards.

Case	Evaluation Metrics		Geometrical Parameters	
	$\overline{x}$ (%)	σ (%)	$\overline{x}$ (%)	σ (%)
A	94.67	1.72	1.93	1.39
B	88.97	3.73	21.06	18.60
C	94.13	1.93	7.99	10.67
D	85.18	4.28	14.91	9.57
E	87.60	4.39	12.60	3.91
F	94.27	2.38	4.89	3.12
G	90.25	4.12	16.77	15.79
H	86.33	7.86	25.90	18.89

4.8 Summary

The analysis of computational efficiency evaluated each method based on both training effort and prediction generation efficiency. The results indicated that the use of pre-trained networks as backbones in the DeepLabv3+ variants significantly reduced training time. Traditional Otsu's segmentation method required minimal computational effort for inference due to its straightforward algorithm. Among the deep learning models, DeepLabv3+ with ResNet-18 and MobileNetV2 backbones demonstrated the best inference performance, with the shortest inference times and highest FPS. In terms of overall crack segmentation accuracy, advanced deep learning models such as Res-UNet and DeepLabv3+ variants consistently achieved high precision, recall,

F1 Score, and IoU, reflecting their superior capability in crack detection and segmentation. DeepLabv3+ variants emerged as the most effective architectures as evidenced by their higher IoU values. The case-wise segmentation performance analysis provided valuable insights into the robustness and adaptability of the models under challenging conditions. Overall, Res-UNet and DeepLabv3+ variants exhibited their effectiveness in handling segmentation tasks under various conditions, with DeepLabv3+ models outperforming Res-UNet in accurately reconstructing crack network representations. In summary, this chapter highlights the importance of selecting the appropriate model based on specific requirements and constraints, such as computational resources and the characteristics of crack images.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

The present study was designed to evaluate the feasibility and effectiveness of image-based techniques for quantifying desiccation cracking behaviour in expansive soils. To achieve this, three research objectives were pursued, and their outcomes are summarised below.

As per objective one, which is to design the experimental setup for desiccation crack image acquisition, an experimental setup was successfully developed using a humidity chamber and glass Petri dish specimens. The setup was used for desiccation tests that enabled systematic generation of soil desiccation cracks. Expansive soil desiccation crack images were acquired using the setup. The images produced a diverse dataset that was used for deep learning model training. This confirmed the practicality of image-based experimental acquisition in soil desiccation cracks study.

Objective two is to evaluate crack imaging analysis based on deep learning algorithms. Crack imaging analysis based on deep learning algorithms was evaluated in two categories, including computational efficiency and segmentation performance. The computational efficiency analysis revealed that DeepLabv3+ variants required significantly less training effort due to the

utilisation of transfer learning. Among these, MobileNetV2 variant, with its minimal parameter count and short training time, emerged as the most efficient model considering computational resources. Traditional Otsu's method, due to its simple computation algorithm, exhibited a major advantage in inference speed. Among deep learning models, DeepLabv3+ with ResNet-18 and MobileNetV2 demonstrated the best computational efficiency, characterised by the shortest inference time.

In terms of segmentation performance, this study demonstrated that advanced deep learning models, such as Res-UNet and DeepLabv3+ variants, excelled in the accurate detection and segmentation of soil cracks. These models consistently achieved high precision, recall, F1 score, and IoU, outperforming traditional segmentation methods like Otsu's algorithm and the simpler U-Net network. Notably, DeepLabv3+ variants emerged as the top performers with the highest IoU values.

Furthermore, the advanced deep learning models exhibited a notable ability to produce accurate representations of crack networks, enabling the precise quantification of crack geometrical parameters through image processing techniques. This capability provided valuable insights into soil desiccation cracking behaviour. In contrast, segmentation methods like Otsu's algorithm and U-Net demonstrated limited performance in quantifying crack geometrical characteristics, as evidenced by their high error rates across all parameters. Although Res-UNet showed significantly lower error rates compared to these methods, it was less effective in reconstructing crack network

connectivity, thus limiting its predictive accuracy. DeepLabv3+ models, however, achieved the highest accuracy in quantifying geometrical parameters, indicating their ability to generate segmentation masks with high resemblance to ground truth.

The stability analysis highlighted that traditional methods, such as Otsu's were highly variable and unreliable, particularly under varying image conditions, as demonstrated in case-wise detection analysis. DeepLabv3+ variants with ResNet-50 and ResNet-18 exhibited the most consistent performance across evaluation metrics, with MobileNetV2 variant followed closely behind. In terms of geometrical parameters quantification, DeepLabv3+ with MobileNetV2 emerged as the top performer with the best scores in both accuracy and stability indicators.

The case-wise segmentation performance analysis further illustrated that DeepLabv3+ models were robust and adaptable under various challenging conditions, such as uneven illumination and the presence of soil clods or shadows. The analysis clearly indicated that traditional segmentation methods, such as Otsu's, were limited in handling complex real-world scenarios.

Overall, objective two confirmed the feasibility and effectiveness of deep learning-based image segmentation models for soil crack recognition and quantification. The findings suggested that DeepLabv3+ variants, particularly those with MobileNetV2 and ResNet-18 backbones, offer the best performance, achieving an optimal balance of computational efficiency, segmentation

accuracy, and stability. The study also provided valuable insights into the integration and implementation of deep learning models in soil crack quantification systems.

Objective three is to quantify desiccation cracking through image processing techniques. In this study, image processing techniques were used to effectively measure crack geometrical parameters, which were crucial for cracking analysis. Parameters such as surface crack ratio, crack widths, number of intersections, number of crack segments, and crack lengths were defined and quantified. This effectively proved the feasibility of using image processing techniques in desiccation cracking quantification.

The findings confirmed that deep learning provides a breakthrough in soil crack analysis. By automating segmentation and quantification, this study demonstrated the feasibility of creating a reproducible and scalable image-based processing pipeline for expansive soil research. For civil engineering practice, the accurate quantification of crack geometry enhances the understanding of desiccation mechanisms, which directly affect the study of slope stability, foundation performance, and any other expansive soil applications. From an AI perspective, the successful application of deep learning models demonstrated in this study highlights the value of deep learning integration into the geotechnical field. In addition, it also paved the way for future field-based crack recognition systems that can operate in real-time.

In summary, this study has achieved all objectives and contributed a validated workflow for using deep learning-based segmentation and image processing techniques in soil desiccation crack recognition and quantification.

5.2 Recommendations for Future Work

Based on the limitations and results of this study, suggestions for future research are proposed. First, further refinement of the deep learning models is essential. This includes exploring different deep learning frameworks and integrating advanced techniques to enhance recognition accuracy. Further research should also address image variability issues by employing techniques such as data augmentation and synthetic data generation. These approaches could help in training more resilient models with improved generalisation capabilities. To broaden the applicability of the models, further studies should consider incorporating additional labels, such as different soil types and environmental conditions, into the image dataset. This expansion may contribute to the development of more robust crack detection models capable of performing well across diverse soil types.

For practical implementations, it is crucial to apply these segmentation models in real-world field conditions to validate their practicality. The development of user-friendly interfaces or mobile applications for on-site crack detection and analysis systems is recommended to enhance system accessibility. Additionally, optimising models for real-time processing and integrating them with automated systems, such as unmanned aerial vehicles, could significantly

advance soil monitoring systems. Exploring and comparing more lightweight model architectures that strike a balance between accuracy and computational demands will also be beneficial in developing efficient real-time applications.

REFERENCES

Albrecht, B.A. and Benson, C.H., 2001. Effect of desiccation on compacted natural clays. *Journal of Geotechnical and Geoenvironmental Engineering*, [e-journal] 127(1), pp.67–75. http://dx.doi.org/10.1061/(ASCE)1090-0241(2001)127:1(67)

Al-Jeznawi, D., Sanchez, M. and Al-Taie, A.J., 2020. Effect of wetting-drying cycles on desiccation crack pattern and soil behavior. *Key Engineering Materials*, [e-journal] 857, pp.188–194. https://doi.org/10.4028/www.scientific.net/KEM.857.188

Al-Jeznawi, D., Sanchez, M. and Al-Taie, A.J., 2021. Using Image Analysis Technique to Study the Effect of Boundary and Environment Conditions on Soil Cracking Mechanism. *Geotechnical and Geological Engineering*, [e-journal] 39, pp.25–36. https://doi.org/10.1007/s10706-020-01376-5

Alom, M.Z., Taha, T.M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M.S., Hasan, M., Van Essen, B.C., Awwal, A.A.S. and Asari, V.K., 2019. A State-of-the-Art Survey on Deep Learning Theory and Architectures. *Electronics*, [e-journal] 8(3), p.292. https://doi.org/10.3390/electronics8030292

Andrushia, A.D., Neebha, T.M., Umadevi, S., Anand, N., Cashell, K.A., 2022. Shrinkage Crack Detection in Expansive Soil using Deep Convolutional Neural Network and Transfer Learning. *KSCE Journal of Civil Engineering*, [e-journal] 26(10), pp.4202-4213. https://doi.org/10.1007/s12205-022-1354-0

Bamgbopa, O.S., 2016. Investigation of Shrinkage and Cracking in Clay Soils under Wetting and Drying Cycles. *International Journal of Engineering Research & Technology (IJERT)*, [e-journal] 5(11), pp.283–320. http://dx.doi.org/10.17577/IJERTV5IS110183

Benjamin, N., 2024. Drowning in the heatwave. *The Star*, [online] 8 August. Available at: < https://www.thestar.com.my/news/nation/2024/08/08/drowning-in-the-heatwave> [Accessed 19 September 2024].

Bronswijk, J.J.B., 1991. Relation between Vertical Soil Movements and Water-Content Changes in Cracking Clays. *Soil Science Society of America Journal*, [e-journal] 55(5), pp.1220–1226. http://dx.doi.org/10.2136/sssaj1991.03615995005500050004x

Budhu, M., 2010. *Soil mechanics and foundations.* 3rd ed., John Wiley and Sons.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F. and Adam, H., 2018. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In: *Proceedings of the European conference on computer vision*

(ECCV). Munich, Germany, 8 – 14 September 2018. pp. 801-818. https://doi.org/10.1007/978-3-030-01234-2_49

Chen, L.-C., Papandreou, G., Schroff, F. and Adam, H., 2017. Rethinking Atrous Convolution for Semantic Image Segmentation. *ArXiv preprint.* https://doi.org/10.48550/arXiv.1706.05587

Cheng, Q., Gu, Y.D., Tang, C.S., Zhang, X.Y. and Shi, B., 2023. Desiccation cracking behaviour of a vegetated soil incorporating planting density. *Canadian Geotechnical Journal*, [e-journal] 61(1), pp.165–173. https://doi.org/10.1139/cgj-2023-0047

Cheng, Q., Tang, C.S., Zeng, H., Zhu, C., An, N. and Shi, B., 2020. Effects of microstructure on desiccation cracking of a compacted soil. *Engineering Geology*, [e-journal] 265, p.105418. https://doi.org/10.1016/j.enggeo.2019.105418

Cheng, Q., Tang, C.S., Xu, D., Zeng, H. and Shi, B., 2021. Water infiltration in a cracked soil considering effect of drying-wetting cycles. *Journal of Hydrology*, [e-journal] 593, p.125640. https://doi.org/10.1016/j.jhydrol.2020.125640

Chollet, F., 2017. Xception: Deep Learning with Depthwise Separable Convolutions. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI, USA, 22-25 July 2017. pp. 1800-1807. https://doi.org/10.1109/CVPR.2017.195

Civitelli, E., Sortino, A., Lapucci, M., Bagattini, F. and Galvan, G., 2023. A Robust Initialization of Residual Blocks for Effective ResNet Training Without Batch Normalization. *IEEE Transactions on Neural Networks and Learning Systems*, [e-journal] 36(1), pp.1947-1952. https://doi.org/10.1109/TNNLS.2023.3325541

Corte, A. and Higashi, A., 1964. *Experimental Research on Desiccation Cracks in Soil*. New Hampshire: U.S. Army Cold Regions Research and Engineering Laboratory.

Costa, S., Kodikara, J. and Shannon, B., 2013. Salient factors controlling desiccation cracking of clay in laboratory experiments. *Geotechnique*, [e-journal] 63(1), pp.18–29. https://doi.org/10.1680/geot.9.P.105

Das, S., Fime, A.A., Siddique, N. and Hashem, M.M.A., 2021. Estimation of Road Boundary for Intelligent Vehicles Based on DeepLabV3+ Architecture. *IEEE Access*, [e-journal] 9, pp.121060–121075. https://doi.org/10.1109/ACCESS.2021.3107353

Dasog, G.S. and Shashidhara, G.B., 1993. Dimension and volume of cracks in a vertisol under different crop covers. *Soil Science*, [e-journal] 156(6), pp.424–428. http://dx.doi.org/10.1097/00010694-199415860-00009

El Abedine, A.Z. and Robinson, G.H., 1971. A study on cracking in some Vertisols of the Sudan. *Geoderma*, [e-journal] 5(3), pp.229-241. https://doi.org/10.1016/0016-7061(71)90012-7

El Hajjar, A., Ouahbi, T., Eid, J., Taibi, S., Bouchemella, S. and Eid, J., 2019. Desiccation and cracking behaviour of clayey soils: Experimental characterization and mechanisms identification. In *17th European Conference on Soil Mechanics and Geotechnical Engineering, ECSMGE 2019 - Proceedings*. Reykjavik, Iceland, 1 – 6 September 2019. http://doi.org/10.32075/17ECSMGE-2019-0998

Fu, J., Yi, X., Wang, G., Mo, L., Wu, P. and Kapula, K.E., 2022. Research on Ground Object Classification Method of High Resolution Remote-Sensing Images Based on Improved DeeplabV3+. *Sensors*, [e-journal] 22(19), p.7477. https://doi.org/10.3390/s22197477

Gao, Q.-F., Zeng, L. and Shi, Z.-N., 2021. Effects of desiccation cracks and vegetation on the shallow stability of a red clay cut slope under rainfall infiltration. *Computers and Geotechnics*, [e-journal] 140, p.104436. https://doi.org/10.1016/j.compgeo.2021.104436

Gayakwad, E., Prabhu, J., Anand, R.V. and Kumar, M.S., 2021. Training Time Reduction in Transfer Learning for a Similar Dataset Using Deep Learning. In: Satapathy, S., Zhang, Y.D., Bhateja, V., Majhi, R. (eds) *Intelligent Data Engineering and Analytics*. Singapore: Springer. pp. 359-367. https://doi.org/10.1007/978-981-15-5679-1_33

Glorot, X. and Bengio, Y., 2010. Understanding the difficulty of training deep feedforward neural networks. In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS) 2010*. Sardinia, Italy, 13-15 May 2010. pp. 249–256.

Gonzalez, R.C. and Woods, R.E., 2008. *Digital Image Processing*. 3rd ed. Upper Saddle River: Pearson Education, Inc.

Goodfellow, I., Bengio, Y. and Courville, A., 2016. *Deep Learning*. Cambridge: MIT Press.

Grauman, K. and Darrell, T., 2005. The pyramid match kernel: discriminative classification with sets of image features. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, pp. 1458–1465. IEEE. http://dx.doi.org/10.1109/ICCV.2005.239

Gu, K., Tang, C.S., Shi, B., Hong, J.J. and Jin, F., 2014. A study of the effect of temperature on the structural strength of a clayey soil using a micropenetrometer. *Bulletin of Engineering Geology and the Environment*, [e-journal] 73(3), pp.747–758. https://doi.org/10.1007/s10064-013-0543-y

Hadinata, P.N., Simanta, D., Eddy, L. and Nagai, K., 2021. Crack Detection on Concrete Surfaces Using Deep Encoder-Decoder Convolutional Neural

Network: A Comparison Study Between U-Net and DeepLabV3+. *Journal of the Civil Engineering Forum*, [e-journal] 7(3), p.323. http://dx.doi.org/10.22146/jcef.65288

Hadjiiski, L., Samala, R. and Chan, H.-P., 2021. Image Processing Analytics: Enhancements and Segmentation. In: Ross, B.D. and Gambhir, S.S., (eds.) *Molecular Imaging: Principles and Practice*. 2nd ed. Elsevier, pp. 1727–1745. https://doi.org/10.1016/B978-0-12-816386-3.00057-0

Han, X.L., Jiang, N.J., Yang, Y.F., Choi, J., Singh, D.N., Beta, P., Du, Y.J. and Wang, Y.J., 2022. Deep learning based approach for the instance segmentation of clayey soil desiccation cracks. *Computers and Geotechnics*, [e-journal] 146(11), p.104733. https://doi.org/10.1016/j.compgeo.2022.104733

He, K., Zhang, X., Ren, S. and Sun, J., 2016a. Deep Residual Learning for Image Recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Nevada, USA, 27-30 June 2016. pp. 770–778. https://doi.org/10.1109/CVPR.2016.90

He, K., Zhang, X., Ren, S. and Sun, J., 2015a. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. Santiago, Chile, 7-13 December 2015. pp. 1026–1034. https://doi.org/10.1109/ICCV.2015.123

He, K., Zhang, X., Ren, S. and Sun, J., 2016b. Identity Mappings in Deep Residual Networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds) *Computer Vision – ECCV 2016*. Cham: Springer. pp.630-645. https://doi.org/10.1007/978-3-319-46493-0_38

He, K., Zhang, X., Ren, S. and Sun, J., 2015b. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, [e-journal] 37(9), pp.1904–1916. https://doi.org/10.1109/TPAMI.2015.2389824

Issam, E.N. and Murphy, M.J., 2015. What Is Machine Learning? In: *Machine Learning in Radiation Oncology.* Cham: Springer International Publishing. pp. 3–11. https://doi.org/10.1007/978-3-319-18305-3_1

Janiesch, C., Zschech, P. and Heinrich, K., 2021. Machine learning and deep learning. *Electronic Markets*, [e-journal] 31(3), pp.685–695. https://doi.org/10.1007/s12525-021-00475-2

Jones, L.D. and Jefferson, I., 2012. Expansive soils. In: Burland, J. (eds.) *ICE manual of geotechnical engineering*. London: ICE Publishing. pp. 413–441.

Kingma, D.P. and Ba, J., 2015. Adam: A Method for Stochastic Optimization. In: *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015).* California, USA, 7-9 May 2015. https://doi.org/10.48550/arXiv.1412.6980

Kodikara, J. and Costa, S., 2013. Desiccation Cracking in Clayey Soils: Mechanisms and Modelling. In: Laloui, L. and Ferrari, A., (eds.) *Multiphysical testing of soils and shales*. Berlin: Springer, Berlin, Heidelberg. pp. 21–32. https://doi.org/10.1007/978-3-642-32492-5_2

Koons, E., 2024. *Climate Change in Malaysia - Environmental Issues and Solutions* [Online]. Available at: <https://energytracker.asia/climate-change-in-malaysia-environmental-issues-and-solutions/> [Accessed: 19 September 2024].

Lazebnik, S., Schmid, C. and Ponce, J., 2006. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. New York, USA, 17-22 June 2006. pp. 2169–2178. https://doi.org/10.1109/CVPR.2006.68

LeCun, Y., Bengio, Y. and Hinton, G., 2015. Deep learning. *Nature*, [e-journal] 521(7553), pp.436–444. https://doi.org/10.1038/nature14539

Li, H.D., Tang, C.S., Cheng, Q., Li, S.J., Gong, X.P. and Shi, B., 2019. Tensile strength of clayey soil and the strain analysis based on image processing techniques. *Engineering Geology*, [e-journal] 253, pp.137–148. https://doi.org/10.1016/j.enggeo.2019.03.017

Lin, T.Y., Goyal, P., Girshick, R., He, K. and Dollar, P., 2018. Focal Loss for Dense Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, [e-journal] 42(2), pp. 318-327. https://doi.org/10.1109/TPAMI.2018.2858826

Liu, C., Tang, C.S., Shi, B. and Suo, W.B., 2013. Automatic quantification of crack patterns by image processing. *Computers and Geosciences*, [e-journal] 57, pp.77–80. https://doi.org/10.1016/j.cageo.2013.04.008

Lu, Y., Liu, S., Weng, L., Wang, L., Li, Z. and Xu, L., 2016. Fractal analysis of cracking in a clayey soil under freeze-thaw cycles. *Engineering Geology*, [e-journal] 208, pp.93–99. https://doi.org/10.1016/j.enggeo.2016.04.023

Miller, C.J., Mi, H. and Yesiller, N., 1998. Experimental Analysis of Desiccation Crack Propagation in Clay Liners. *Journal of the American Water Resources Association*, [e-journal] 34(3), pp.677-686. https://doi.org/10.1111/j.1752-1688.1998.tb00964.x

Mohan, A. and Poobal, S., 2018. Crack detection using image processing: A critical review and analysis. *Alexandria Engineering Journal*, [e-journal] 57(2), pp.787–798. https://doi.org/10.1016/j.aej.2017.01.020

Morris, H., Graham, J. and Williams, D.J., 1992. Cracking in drying soils. *Canadian Geotechnical Journal*, [e-journal] 29(2), pp.263–277. https://doi.org/10.1139/t92-030

Munawar, H.S., Hammad, A.W.A., Haddad, A., Soares, C.A.P. and Waller, S.T., 2021. Image-based crack detection methods: A review. *Infrastructures*, [e-journal] 6(8), p.115. https://doi.org/10.3390/infrastructures6080115

Murphy, K.P., 2012. *Machine Learning: A Probabilistic Perspective*. Cambridge: MIT Press.

Nahlawi, H. and Kodikara, J.K., 2006. Laboratory experiments on desiccation cracking of thin soil layers. *Geotechnical and Geological Engineering*, [e-journal] 24(6), pp.1641–1664. https://doi.org/10.1007/s10706-005-4894-4

O'Shea, K. and Nash, R., 2015. An Introduction to Convolutional Neural Networks. *ArXiv preprint*. https://doi.org/10.48550/arXiv.1511.08458

Peng, X., Horn, R., Peth, S. and Smucker, A., 2006. Quantification of soil shrinkage in 2D by digital image processing of soil surface. *Soil and Tillage Research*, [e-journal] 91(1–2), pp.173–180. https://doi.org/10.1016/j.still.2005.12.012

Pham, M.V., Ha, Y.S. and Kim, Y.T., 2023. Automatic detection and measurement of ground crack propagation using deep learning networks and an image processing technique. *Measurement*, [e-journal] 215, p.112832. https://doi.org/10.1016/j.measurement.2023.112832

Pisner, D.A. and Schnyer, D.M., 2020. Support vector machine. In: Mechelli, A., Vieira, S. (eds.) *Machine Learning: Methods and Applications to Brain Disorders*. Elsevier, pp. 101–121. https://doi.org/10.1016/B978-0-12-815739-8.00006-7

Prat, P.C., Ledesma, A. and Lakshmikantha, M.R., 2006. Size Effect in the Cracking of Drying Soil. In: Gdoutos, E.E. (eds.) *Fracture of Nano and Engineering Materials and Structures*. Dordrecht: Springer Netherlands. pp. 1373–1374. https://doi.org/10.1007/1-4020-4972-2_682

Rayhani, M.H.T., Yanful, E.K. and Fakher, A., 2008. Physical modeling of desiccation cracking in plastic soils. *Engineering Geology*, [e-journal] 97(1–2), pp.25–31. https://doi.org/10.1016/j.enggeo.2007.11.003

Ronneberger, O., Fischer, P. and Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W., Frangi, A. (eds.) *Medical image computing and computer-assisted intervention-MICCAI 2015*. Cham: Springer. pp. 234–241. https://doi.org/10.1007/978-3-319-24574-4_28

Salimi, K., Cerato, A.B., Vahedifard, F. and Miller, G.A., 2021. Tensile Strength of Compacted Clays during Desiccation under Elevated Temperatures. *Geotechnical Testing Journal*, [e-journal] 44(4), pp.1119–1134. https://doi.org/10.1520/GTJ20200114

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. and Chen, L.-C., 2018. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Utah, USA, 18-23 June 2018. pp. 4510–4520. https://doi.org/10.1109/CVPR.2018.00474

Shepidchenko, T., Zhang, J., Tang, X., Liu, T., Dong, Z., Zheng, G. and Yang, L., 2020. Experimental study of the main controlling factors of desiccation crack formation from mud to shale. *Journal of Petroleum Science and Engineering*, [e-journal] 194, p.107414. https://doi.org/10.1016/j.petrol.2020.107414

Shit, P.K., Bhunia, G.S. and Maiti, R., 2015. Soil crack morphology analysis using image processing techniques. *Modeling Earth Systems and Environment*, [e-journal] 1(4), pp.1–7. https://doi.org/10.1007/s40808-015-0036-z

Shorten, C. and Khoshgoftaar, T.M., 2019. A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, [e-journal] 6(1), pp.1–48. https://doi.org/10.1186/s40537-019-0197-0

Singh, S.P., Rout, S. and Tiwari, A., 2018. Quantification of desiccation cracks using image analysis technique. *International Journal of Geotechnical Engineering*, [e-journal] 12(4), pp.383–388. https://doi.org/10.1080/19386362.2017.1282400

Solomon, C. and Breckon, T., 2011. *Fundamentals of Digital Image Processing: A practical approach with examples in Matlab*. John Wiley & Sons.

Tan, M.L., Juneng, L., Tangang, F.T., Chung, J.X. and Firdaus, R.B.R., 2021. Changes in temperature extremes and their relationship with ENSO in Malaysia from 1985 to 2018. *International Journal of Climatology*, [e-journal] 41(S1), pp.E2564–E2580. http://dx.doi.org/10.1002/joc.6864

Tang, C.S., Shi, B., Liu, C., Zhao, L.Z., Wang, B.J., 2008. Influencing factors of geometrical structure of surface shrinkage cracks in clayey soils. *Engineering Geology*, [e-journal] 101(3–4), pp.204–217. https://doi.org/10.1016/j.enggeo.2008.05.005

Tang, C.S., Cui, Y.J., Shi, B., Tang, A.M. and Liu, C., 2011a. Desiccation and cracking behaviour of clay layer from slurry state under wetting-drying cycles. *Geoderma*, [e-journal] 166(1), pp.111–118. https://doi.org/10.1016/j.geoderma.2011.07.018

Tang, C.S., Zhu, C., Cheng, Q., Zeng, H., Xu, J.J., Tian, B.G. and Shi, B., 2021. Desiccation cracking of soils: A review of investigation approaches, underlying mechanisms, and influencing factors. *Earth-Science Reviews*, [e-journal] 216, p.103586. https://doi.org/10.1016/j.earscirev.2021.103586

Tang, C.S., Shi, B., Liu, C., Suo, W.B. and Gao, L., 2011b. Experimental characterization of shrinkage and desiccation cracking in thin clay layer.

Applied Clay Science, [e-journal] 52(1–2), pp.69–77. https://doi.org/10.1016/j.clay.2011.01.032

Tang, C.S., Cui, Y.J., Tang, A.M. and Shi, B., 2010. Experiment evidence on the temperature dependence of desiccation cracking behavior of clayey soils. *Engineering Geology*, [e-journal] 114(3–4), pp.261–266. https://doi.org/10.1016/j.enggeo.2010.05.003

Tao, P., Yi, H., Wei, C., Ge, L.Y. and Xu, L., 2013. A method based on weighted F-score and SVM for feature selection. In: *2013 25th Chinese Control and Decision Conference (CCDC)*. Guiyang, China, 25-27 May 2013. pp. 4287–4290. https://doi.org/10.1109/CCDC.2013.6561705

The Star, 2023. Asia's heatwave could get worse with El Nino. *The Star*, [online] 30 May. Available at: < https://www.thestar.com.my/news/nation/2023/05/30/asias-heatwave-could-get-worse-with-el-nino> [Accessed 19 September 2024].

Vogel, H.J., Hoffmann, H. and Roth, K., 2005. Studies of crack dynamics in clay soil: I. Experimental methods, results, and morphological quantification. *Geoderma*, [e-journal] 125(3–4), pp.203–211. https://doi.org/10.1016/j.geoderma.2004.07.009

Wang, L.L., Tang, C.S., Shi, B., Cui, Y.J., Zhang, G.Q. and Hilary, I., 2018. Nucleation and propagation mechanisms of soil desiccation cracks. *Engineering Geology*, [e-journal] 238, pp.27–35. https://doi.org/10.1016/j.enggeo.2018.03.004

Wang, P., Fan, E. and Wang, P., 2021. Comparative analysis of image classification algorithms based on traditional machine learning and deep learning. *Pattern Recognition Letters*, [e-journal] 141, pp.61–67. https://doi.org/10.1016/j.patrec.2020.07.042

Wei, X., Hattab, M., Bompard, P. and Fleureau, J.-M., 2016. Highlighting some mechanisms of crack formation and propagation in clays on drying path. *Geotechnique*, [e-journal] 66(4), pp.287-300. https://doi.org/10.1680/jgeot.14.P.227

Xin, Y., Kong, L., Liu, Z., Chen, Y., Li, Y. and Zhu, H., 2018. Machine Learning and Deep Learning Methods for Cybersecurity. *IEEE Access*, [e-journal] 6, pp.35365–35381. https://doi.org/10.1109/ACCESS.2018.2836950

Xu, J.J., Zhang, H., Tang, C.S., Cheng, Q., Liu, B. and Shi, B., 2022a. Automatic soil desiccation crack recognition using deep learning. *Geotechnique*, [e-journal] 72(4), pp.337–349. https://doi.org/10.1680/jgeot.20.P.091

Xu, J.J., Zhang, H., Tang, C.S., Cheng, Q., Tian, B.G., Liu, B. and Shi, B., 2022b. Automatic soil crack recognition under uneven illumination condition

with the application of artificial intelligence. *Engineering Geology*, [e-journal] 296, p.106495. https://doi.org/10.1016/j.enggeo.2021.106495

Xu, J.J., Zhang, H., Tang, C.S., Yang, Y., Li, L., Wang, D.L., Liu, B., Shi, B., 2024. Soil Desiccation Crack Recognition: New Paradigm and Field Application. *Journal of Geophysical Research: Machine Learning and Computation*, [e-journal] 1(3), p. e2024JH000347. https://doi.org/10.1029/2024JH000347

Yang, B., Yuan, S., Shen, Z. and Zhao, X., 2022. Influence of Geotextile Materials on the Fractal Characteristics of Desiccation Cracking of Soil. *Fractal and Fractional*, [e-journal] 6(11), p.628. https://doi.org/10.3390/fractalfract6110628

Yesiller, N., Miller, C.J., Inci, G. and Yaldo, K., 2000. Desiccation and cracking behavior of three compacted landfill liner soils. *Engineering Geology*, [e-journal] 57(1–2), pp.105–121. https://doi.org/10.1016/S0013-7952(00)00022-3

Yu, W.-W., He, F. and Xi, P., 2010. A rapid 3D seed-filling algorithm based on scan slice. *Computers & Graphics*, [e-journal] 34(4), pp.449–459. https://doi.org/10.1016/j.cag.2010.05.005

Zeng, H., Tang, C.S., Cheng, Q., Inyang, H.I., Rong, D.Z., Lin, L. and Shi, B., 2019a. Coupling effects of interfacial friction and layer thickness on soil desiccation cracking behavior. *Engineering Geology*, [e-journal] 260, p.105220. https://doi.org/10.1016/j.enggeo.2019.105220

Zeng, H., Tang, C.S., Cheng, Q., Lin, L. and Xu, J.J., 2019b. Desiccation cracking behavior of soils. *7th Asia-Pacific Conference on Unsaturated Soils (AP-UNSAT 2019), 2019 Japanese Geotechnical Society Special Publication*, [e-journal] 7(2), pp. 90–95. https://doi.org/10.3208/jgssp.v07.013

Zeng, H., Tang, C.S., Zhu, C., Vahedifard, F., Cheng, Q. and Shi, B., 2022. Desiccation cracking of soil subjected to different environmental relative humidity conditions. *Engineering Geology*, [e-journal] 297, p.106536. https://doi.org/10.1016/j.enggeo.2022.106536

Zhang, F., Hu, Z., Yang, K., Fu, Y., Feng, Z. and Bai, M., 2021. The surface crack extraction method based on machine learning of image and quantitative feature information acquisition method. *Remote Sensing*, [e-journal] 13(8), p.1534. https://doi.org/10.3390/rs13081534

Zhang, Z., Liu, Q. and Wang, Y., 2018. Road Extraction by Deep Residual U-Net. *IEEE Geoscience and Remote Sensing Letters*, [e-journal] 15(5), pp.749–753. https://doi.org/10.1109/LGRS.2018.2802944

Zhao, C. and Koseki, J., 2020. An image-based method for evaluating local deformations of saturated sand in undrained torsional shear tests. *Soils and*

Foundations, [e-journal] 60(3), pp.608-620.
https://doi.org/10.1016/j.sandf.2020.02.012

APPENDICES

Appendix A

Temperature and Relative Humidity Records for Soil Desiccation Tests

Table A.1: Test condition – 5mm specimen, 45 ± 2°C, 40 ± 5% RH

Cycle 1			Cycle 2		
Time	**Temperature (°C)**	**Humidity (%)**	**Time**	**Temperature (°C)**	**Humidity (%)**
7/5/22 16:00	46.4	35.5	7/7/22 17:00	46.6	36.1
7/5/22 19:00	46.2	39.3	7/7/22 20:00	46.3	37.5
7/5/22 22:00	46.8	35.2	7/7/22 23:00	46.0	37.2
7/5/22 23:00	45.0	38.5	7/7/22 23:30	45.4	35.5
7/6/22 3:00	43.1	42.8	7/8/22 3:30	43.8	41.8
7/6/22 7:00	44.0	42.4	7/8/22 5:30	43.5	40.3
7/6/22 10:00	44.6	41.7	7/8/22 7:30	44.1	41.1
7/6/22 13:00	46.5	40.3	7/8/22 10:30	46.1	39.1
7/6/22 16:00	46.6	40.1			
7/6/22 19:00	46.9	40.3			
7/6/22 21:00	45.4	38.9			

Table A.1 (cont)

Cycle 3			Cycle 4		
Time	**Temperature (°C)**	**Humidity (%)**	**Time**	**Temperature (°C)**	**Humidity (%)**
7/10/22 11:00	44.2	42.8	7/13/22 10:00	44.9	41.6
7/10/22 13:00	46.9	41.5	7/13/22 13:00	45.4	37.6
7/10/22 16:00	46.1	36.3	7/13/22 16:00	46.8	37.9
7/10/22 18:00	46.9	39.4	7/13/22 19:00	46.3	38.2
7/10/22 21:00	46.4	35.8	7/13/22 21:00	46.7	36.6
7/11/22 0:00	45.8	36.1	7/14/22 1:00	45.0	37.5
7/11/22 3:00	44.5	40.0	7/14/22 5:00	45.6	43.7
7/11/22 5:30	43.1	42.3	7/14/22 8:00	44.7	41.3
7/11/22 10:00	44.1	42.6	7/14/22 11:00	44.8	42.3
7/11/22 13:00	45.0	38.6	7/14/22 13:00	46.0	38.1
			7/14/22 16:00	45.5	39.0

Table A.2: Test condition – 10mm specimen, 45 ± 2°C, 40 ± 5% RH

Cycle 1			Cycle 2		
Time	**Temperature (°C)**	**Humidity (%)**	**Time**	**Temperature (°C)**	**Humidity (%)**
7/5/22 16:00	46.3	38.1	7/8/22 23:30	46.0	35.7
7/5/22 19:00	46.5	39.1	7/9/22 0:30	45.6	38.7
7/5/22 22:00	46.4	39.6	7/9/22 3:30	44.5	35.9
7/5/22 23:00	45.8	35.4	7/9/22 6:30	44.1	36.5
7/6/22 3:00	45.0	42.6	7/9/22 10:00	44.9	42.3
7/6/22 7:00	43.3	41.4	7/9/22 13:30	45.8	43.3
7/6/22 10:00	44.4	42.9	7/9/22 17:00	45.0	42.7
7/6/22 13:00	46.7	37.1	7/9/22 20:00	46.4	40.8
7/6/22 16:00	45.3	39.9	7/10/22 0:00	45.0	39.6
7/6/22 19:00	45.6	38.9	7/10/22 3:00	43.6	43.2
7/6/22 22:00	46.0	37.4	7/10/22 7:00	43.7	43.5
7/7/2022 0:00	44.3	43.3	7/10/22 10:00	44.9	41.2
7/7/2022 4:00	44.8	44.9	7/10/22 13:00	45.6	36.3
7/7/2022 7:00	44.0	42.1	7/10/22 16:00	45.8	38.8
7/7/2022 10:00	43.8	40.6			
7/7/2022 13:00	45.1	37.4			
7/7/2022 17:00	44.8	37.7			

Table A.2 (cont)

Cycle 3			Cycle 4		
Time	**Temperature (°C)**	**Humidity (%)**	**Time**	**Temperature (°C)**	**Humidity (%)**
7/13/22 10:00	44.1	41.3	7/16/22 21:00	45.0	37.0
7/13/22 13:00	46.5	39.1	7/16/22 23:00	44.8	36.5
7/13/22 16:00	46.6	35.7	7/17/22 3:00	44.1	42.7
7/13/22 19:00	47.0	36.4	7/17/22 6:00	44.9	40.9
7/13/22 21:00	46.3	36.0	7/17/22 9:00	45.2	41.8
7/14/22 1:00	45.0	39.6	7/17/22 12:00	46.0	41.9
7/14/22 5:00	43.2	44.8	7/17/22 14:00	46.3	43.9
7/14/22 8:00	44.8	42.5	7/17/22 17:00	45.2	42.7
7/14/22 11:00	44.9	42.6	7/17/22 20:00	45.0	38.6
7/14/22 13:00	45.4	37.8	7/17/22 23:00	45.5	39.7
7/14/22 16:00	45.9	42.0	7/18/22 0:30	45.7	35.3
7/14/22 19:00	43.1	43.5	7/18/22 3:30	44.4	41.4
7/14/22 20:00	43.3	43.0	7/18/22 5:30	43.4	43.2
7/15/22 0:00	43.4	42.6	7/18/22 8:30	44.6	41.3
7/15/22 1:30	45.5	39.2	7/18/22 11:30	44.6	42.3
7/15/22 5:30	45.0	37.2	7/18/22 14:30	45.0	38.4
7/15/22 8:30	44.8	43.0	7/18/22 17:30	46.2	38.9
7/15/22 11:30	45.2	43.0	7/18/22 19:30	43.3	44.6
7/15/22 15:00	46.9	42.1	7/18/22 21:30	44.4	42.7

Table A.3: Test condition – 5mm specimen, 45 ± 2°C, 60 ± 5% RH

Cycle 1			Cycle 2			Cycle 3		
Time	**Temperature (°C)**	**Humidity (%)**	**Time**	**Temperature (°C)**	**Humidity (%)**	**Time**	**Temperature (°C)**	**Humidity (%)**
2/28/23 14:30	43.6	57.0	3/3/23 17:45	46.7	55.4	3/6/23 18:30	43.5	60.8
2/28/23 15:00	43.6	56.9	3/3/23 18:00	45.6	56.8	3/6/23 19:00	43.8	61.4
2/28/23 15:30	43.9	56.7	3/3/23 18:30	45.6	60.2	3/6/23 19:30	44.4	61.1
2/28/23 16:00	43.9	58.4	3/3/23 19:30	45.5	64.2	3/6/23 20:00	43.2	62.1
2/28/23 16:30	43.7	60.6	3/3/23 20:30	45.6	64.6	3/6/23 22:00	43.5	62.3
2/28/23 18:00	43.8	61.4	3/3/23 21:30	45.3	64.9	3/7/23 0:30	44.1	64.8
2/28/23 19:00	43.6	61.7	3/3/23 22:30	45.5	64.8	3/7/23 1:00	44.1	64.3
2/28/23 19:30	43.3	61.6	3/4/23 0:30	45.6	64.6	3/7/23 3:00	44.3	64.3
2/28/23 21:30	43.7	63.8	3/4/23 2:30	46.0	64.4	3/7/23 5:00	44.5	64.1
2/28/23 23:30	43.8	62.5	3/4/23 3:30	46.0	64.5	3/7/23 8:30	44.8	65.0
3/1/23 1:00	43.4	62.4	3/4/23 6:30	46.2	64.1	3/7/23 9:00	44.8	64.7
3/1/23 4:00	43.0	62.5	3/4/23 9:30	46.1	64.7	3/7/23 11:00	44.3	64.3
3/1/23 7:00	43.6	62.8	3/4/23 11:30	46.4	63.7	3/7/23 13:00	44.4	64.7
3/1/23 9:00	43.0	62.7	3/4/23 14:30	46.7	63.3	3/7/23 15:00	44.5	64.7
3/1/23 11:00	43.1	63.1	3/4/23 17:30	46.7	63.7	3/7/23 17:30	44.5	64.6
3/1/23 12:00	44.5	62.9	3/4/23 19:30	46.3	62.6	3/7/23 19:30	44.1	64.8
3/1/23 13:00	44.7	63.8	3/4/23 21:30	46.0	64.1	3/7/23 21:30	44.6	63.6
3/1/23 15:00	44.8	63.3	3/4/23 22:30	46.1	63.8	3/7/23 23:30	44.8	61.5
3/1/23 17:00	44.8	63.0	3/4/23 23:30	46.6	64.6	3/8/23 1:30	44.8	60.8

Table A.3 (cont.)

Cycle 1			Cycle 2			Cycle 3		
Time	**Temperature (°C)**	**Humidity (%)**	**Time**	**Temperature (°C)**	**Humidity (%)**	**Time**	**Temperature (°C)**	**Humidity (%)**
3/1/23 19:00	44.8	62.3	3/5/23 1:30	46.6	62.0	3/8/23 2:30	43.2	60.7
3/1/23 19:30	44.4	62.3	3/5/23 2:30	46.4	58.5	3/8/23 4:30	43.4	58.7
3/1/23 21:30	43.6	61.5	3/5/23 5:30	46.3	57.6	3/8/23 6:30	43.6	56.9
3/1/23 23:00	44.4	62.9	3/5/23 8:30	46.4	56.2	3/8/23 8:30	43.5	55.1
3/2/23 1:00	44.7	61.4	3/5/23 10:00	46.5	55.7	3/8/23 10:30	44.0	55.4
3/2/23 2:30	44.6	56.7	3/5/23 12:00	46.6	55.2	3/8/23 11:30	43.7	55.2
3/2/23 4:30	44.7	56.3	3/5/23 14:00	46.7	56.3	3/8/23 13:30	44.0	55.1
3/2/23 6:30	44.9	56.5	3/5/23 16:00	43.2	56.5	3/8/23 14:30	43.8	55.5
3/2/23 8:30	44.8	56.8	3/5/23 17:00	46.9	55.5	3/8/23 15:30	43.7	55.3
3/2/23 10:30	44.8	56.9	3/5/23 18:00	46.8	55.8	3/8/23 16:30	43.7	55.2
3/2/23 11:30	44.6	55.1						
3/2/23 12:30	44.7	55.3						
3/2/23 13:30	44.6	55.2						
3/2/23 14:30	44.8	56.9						
3/2/23 15:30	44.5	55.5						

Table A.4: Test condition – 10mm specimen, 45 ± 2°C, 60 ± 5% RH

Cycle 1			Cycle 2			Cycle 3		
Time	**Temperature (°C)**	**Humidity (%)**	**Time**	**Temperature (°C)**	**Humidity (%)**	**Time**	**Temperature (°C)**	**Humidity (%)**
2/11/23 0:30	46.5	65.0	2/14/23 20:45	43.3	59.2	2/18/23 19:15	43.7	55.0
2/11/23 1:30	46.4	64.1	2/14/23 21:00	44.4	62.4	2/18/23 19:30	43.8	57.1
2/11/23 2:30	46.3	64.3	2/14/23 21:30	43.0	62.5	2/18/23 19:45	44.5	56.0
2/11/23 5:30	46.5	64.5	2/14/23 22:00	43.8	63.3	2/18/23 20:00	44.1	55.3
2/11/23 8:30	45.7	65.0	2/14/23 22:30	43.6	64.7	2/18/23 20:15	44.7	57.0
2/11/23 10:30	45.9	65.0	2/14/23 23:30	44.7	64.6	2/18/23 20:45	43.9	59.0
2/11/23 12:30	45.7	64.2	2/15/23 0:30	44.9	65.0	2/18/23 21:15	44.3	61.4
2/11/23 14:30	45.6	64.2	2/15/23 1:30	45.2	64.7	2/18/23 21:45	44.8	62.1
2/11/23 17:00	45.9	64.7	2/15/23 2:30	45.4	63.4	2/18/23 22:15	45.0	62.7
2/11/23 18:00	46.1	64.1	2/15/23 5:30	45.3	55.5	2/18/23 23:15	45.6	63.0
2/11/23 19:00	46.2	64.5	2/15/23 8:30	45.6	58.9	2/19/23 0:15	45.9	63.1
2/11/23 20:00	46.2	64.3	2/15/23 10:00	44.9	63.8	2/19/23 1:15	46.0	63.5
2/11/23 22:00	45.7	65.0	2/15/23 11:30	43.9	55.3	2/19/23 2:15	45.8	64.3
2/11/23 23:00	45.8	45.8	2/15/23 12:30	44.8	63.5	2/19/23 3:15	45.9	63.5
2/12/23 0:00	45.4	45.4	2/15/23 14:30	44.7	64.6	2/19/23 6:15	45.7	63.4
2/12/23 1:00	45.7	45.7	2/15/23 16:30	44.9	62.1	2/19/23 8:15	46.2	64.6
2/12/23 2:00	45.7	45.7	2/15/23 19:00	44.9	63.0	2/19/23 11:15	45.4	64.2
2/12/23 5:00	45.7	64.2	2/15/23 19:30	44.1	62.5	2/19/23 12:15	45.5	65.0
2/12/23 7:00	45.3	64.1	2/15/23 21:30	45.6	61.5	2/19/23 14:15	45.9	64.9

Table A.4 (cont.)

Cycle 1			Cycle 2			Cycle 3		
Time	**Temperature (°C)**	**Humidity (%)**	**Time**	**Temperature (°C)**	**Humidity (%)**	**Time**	**Temperature (°C)**	**Humidity (%)**
2/12/23 9:00	45.5	64.8	2/15/23 23:30	44.8	56.8	2/19/23 16:15	46.5	63.5
2/12/23 11:00	44.8	64.7	2/16/23 1:30	45.1	55.8	2/19/23 17:15	46.5	62.5
2/12/23 12:00	45.2	45.2	2/16/23 2:30	45.5	55.2	2/19/23 18:15	46.1	63.1
2/12/23 13:00	45.6	45.6	2/16/23 5:30	46.0	56.7	2/19/23 21:15	45.9	62.2
2/12/23 14:00	45.3	45.3	2/16/23 8:30	45.8	60.6	2/19/23 23:15	46.1	61.4
2/12/23 16:00	46.3	46.3	2/16/23 10:30	46.0	62.1	2/20/23 0:15	46.0	57.6
2/12/23 18:00	46.1	46.1	2/16/23 11:30	44.2	59.4	2/20/23 1:15	46.1	59.7
2/12/23 20:00	45.5	45.5	2/16/23 12:30	45.3	59.0	2/20/23 2:15	46.3	59.0
2/12/23 22:00	45.7	45.7	2/16/23 13:30	45.8	59.3	2/20/23 3:15	46.3	58.9
2/13/23 0:00	45.1	45.1	2/16/23 14:30	45.9	59.2	2/20/23 6:15	46.6	60.6
2/13/23 2:00	44.8	44.8	2/16/23 17:15	45.0	53.1	2/20/23 8:15	46.6	59.9
2/13/23 5:00	46.1	55.6	2/16/23 19:00	45.6	55.7	2/20/23 10:15	46.5	59.9
2/13/23 8:00	45.8	57.6	2/16/23 20:30	44.8	55.4	2/20/23 12:15	46.0	61.3
2/13/23 10:00	45.3	56.7	2/16/23 21:30	45.7	58.4	2/20/23 14:15	45.8	59.4
2/13/23 11:00	45.6	55.4	2/16/23 22:30	45.7	57.9	2/20/23 16:15	45.5	59.6
2/13/23 12:00	46.1	55.6	2/16/23 23:30	46.6	55.9	2/20/23 17:15	45.9	59.6
2/13/23 13:00	45.7	55.4	2/17/23 0:30	46.5	58.6	2/20/23 18:15	45.9	59.4
			2/17/23 1:30	45.8	56.2	2/20/23 19:15	45.6	59.2
			2/17/23 4:30	47.0	56.7	2/20/23 22:15	46.3	57.9

Table A.4 (cont.)

Cycle 1			Cycle 2			Cycle 3		
Time	**Temperature (°C)**	**Humidity (%)**	**Time**	**Temperature (°C)**	**Humidity (%)**	**Time**	**Temperature (°C)**	**Humidity (%)**
			2/17/23 6:30	44.6	58.6	2/20/23 23:15	46.4	57.3
			2/17/23 7:30	45.8	55.2	2/21/23 0:15	46.6	56.4
			2/17/23 9:30	47.0	59.3	2/21/23 1:15	46.2	56.0
			2/17/23 10:30	46.7	58.4	2/21/23 2:15	47.0	55.0
			2/17/23 11:30	46.4	57.4	2/21/23 4:15	47.0	58.7
			2/17/23 12:30	46.5	57.8	2/21/23 6:15	47.0	58.5
						2/21/23 7:15	46.9	56.6
						2/21/23 9:15	47.0	56.5

Table A.5: Test condition – 15mm specimen, 45 ± 2°C, 60 ± 5% RH

Cycle 1			Cycle 2			Cycle 3		
Time	Temperature (°C)	Humidity (%)	Time	Temperature (°C)	Humidity (%)	Time	Temperature (°C)	Humidity (%)
2/10/23 14:45	46.3	56.0	2/14/23 20:45	43.1	55.0	2/18/23 19:15	43.2	55.6
2/10/23 15:00	45.9	55.5	2/14/23 21:00	44.1	55.3	2/18/23 19:30	43.2	56.0
2/10/23 15:15	45.6	55.5	2/14/23 21:30	43.3	57.2	2/18/23 19:45	43.8	55.5
2/10/23 15:30	45.7	57.8	2/14/23 22:00	43.9	58.6	2/18/23 20:00	43.5	55.0
2/10/23 15:45	45.9	58.5	2/14/23 22:30	43.8	60.2	2/18/23 20:15	43.0	58.5
2/10/23 16:00	45.9	61.3	2/14/23 23:30	44.9	62.9	2/18/23 20:45	43.8	60.1
2/10/23 16:30	45.1	62.9	2/15/23 0:30	45.1	62.9	2/18/23 21:15	44.4	61.2
2/10/23 17:00	45.1	62.5	2/15/23 1:30	45.4	63.1	2/18/23 21:45	44.9	63.5
2/10/23 17:30	45.3	63.1	2/15/23 2:30	45.6	63.5	2/18/23 22:15	45.1	62.9
2/10/23 18:30	45.6	63.5	2/15/23 5:30	45.6	64.7	2/18/23 23:15	45.5	62.8
2/10/23 19:00	45.8	64.5	2/15/23 8:30	45.6	64.7	2/19/23 0:15	45.7	63.4
2/10/23 19:30	45.5	64.5	2/15/23 10:00	45.3	64.6	2/19/23 1:15	45.8	64.2
2/10/23 20:30	45.6	64.8	2/15/23 11:30	44.9	64.5	2/19/23 2:15	46.0	64.4
2/10/23 21:30	46.0	64.8	2/15/23 12:30	45.0	64.5	2/19/23 3:15	46.1	64.7
2/10/23 22:30	46.2	64.7	2/15/23 14:30	44.7	64.9	2/19/23 6:15	46.1	64.6
2/10/23 23:30	46.2	64.4	2/15/23 16:30	44.8	65.0	2/19/23 8:15	46.1	64.6
2/11/23 0:30	46.5	64.9	2/15/23 19:00	44.9	64.1	2/19/23 11:15	45.3	64.5
2/11/23 1:30	46.4	64.7	2/15/23 19:30	45.0	64.3	2/19/23 12:15	45.6	64.4
2/11/23 2:30	46.1	64.7	2/15/23 21:30	46.2	64.7	2/19/23 14:15	46.2	64.4

Table A.5 (cont.)

Cycle 1			Cycle 2			Cycle 3		
Time	Temperature (°C)	Humidity (%)	Time	Temperature (°C)	Humidity (%)	Time	Temperature (°C)	Humidity (%)
2/11/23 5:30	46.3	64.6	2/15/23 23:30	46.5	64.7	2/19/23 16:15	46.5	64.3
2/11/23 8:30	45.6	63.5	2/16/23 1:30	46.5	64.7	2/19/23 17:15	46.3	64.3
2/11/23 10:30	45.3	63.6	2/16/23 2:30	45.9	64.6	2/19/23 18:15	46.3	64.4
2/11/23 12:30	45.3	63.7	2/16/23 5:30	45.8	64.7	2/19/23 21:15	46.0	64.3
2/11/23 14:30	45.3	64.4	2/16/23 8:30	45.9	64.8	2/19/23 23:15	46.0	64.4
2/11/23 17:00	45.7	64.7	2/16/23 10:30	45.2	64.7	2/20/23 0:15	45.8	64.7
2/11/23 18:00	45.9	64.6	2/16/23 11:30	45.1	64.4	2/20/23 1:15	46.0	64.7
2/11/23 19:00	45.9	64.4	2/16/23 12:30	45.6	64.1	2/20/23 2:15	46.2	64.7
2/11/23 20:00	45.6	64.3	2/16/23 13:30	45.9	64.3	2/20/23 3:15	46.3	64.7
2/11/23 22:00	45.9	64.2	2/16/23 14:30	45.9	64.4	2/20/23 6:15	46.2	64.5
2/11/23 23:00	45.9	64.8	2/16/23 17:15	46.5	64.7	2/20/23 8:15	46.0	64.1
2/12/23 0:00	45.8	64.7	2/16/23 19:00	45.8	64.8	2/20/23 10:15	45.8	64.5
2/12/23 1:00	45.7	64.7	2/16/23 20:30	45.3	64.8	2/20/23 12:15	46.0	64.8
2/12/23 2:00	45.8	64.6	2/16/23 21:30	45.8	64.9	2/20/23 14:15	45.5	64.4
2/12/23 5:00	46.1	64.6	2/16/23 22:30	46.6	64.7	2/20/23 16:15	46.1	64.5
2/12/23 7:00	45.7	64.4	2/16/23 23:30	46.6	64.5	2/20/23 17:15	45.8	64.4
2/12/23 9:00	45.8	64.4	2/17/23 0:30	46.8	64.3	2/20/23 18:15	45.8	64.6
2/12/23 11:00	45.2	64.5	2/17/23 1:30	46.9	63.7	2/20/23 19:15	46.2	64.6
2/12/23 12:00	45.1	64.3	2/17/23 4:30	46.1	64.1	2/20/23 22:15	46.2	64.6

Table A.5 (cont.)

Cycle 1			Cycle 2			Cycle 3		
Time	Temperature (°C)	Humidity (%)	Time	Temperature (°C)	Humidity (%)	Time	Temperature (°C)	Humidity (%)
2/12/23 13:00	45.3	64.2	2/17/23 6:30	46.5	64.5	2/20/23 23:15	46.5	64.4
2/12/23 14:00	45.6	64.3	2/17/23 7:30	47.0	64.2	2/21/23 0:15	45.6	64.8
2/12/23 16:00	45.3	64.3	2/17/23 9:30	46.7	64.3	2/21/23 1:15	45.4	64.7
2/12/23 18:00	45.9	64.4	2/17/23 10:30	46.5	64.3	2/21/23 2:15	45.6	64.9
2/12/23 20:00	45.1	64.4	2/17/23 11:30	46.4	64.5	2/21/23 4:15	46.2	64.9
2/12/23 22:00	45.4	63.1	2/17/23 12:30	46.7	64.5	2/21/23 6:15	45.8	65.0
2/13/23 0:00	45.4	62.8	2/17/23 13:30	46.6	64.5	2/21/23 7:15	45.3	63.5
2/13/23 2:00	45.5	59.6	2/17/23 14:00	46.3	64.5	2/21/23 9:15	45.6	63.1
2/13/23 5:00	46.4	56.5	2/17/23 15:00	46.5	64.6	2/21/23 10:15	45.7	62.9
2/13/23 8:00	46.0	55.4	2/17/23 16:00	45.9	64.7	2/21/23 11:15	45.7	61.8
2/13/23 10:00	45.8	55.9	2/17/23 17:00	46.2	64.6	2/21/23 12:15	45.8	60.6
2/13/23 11:00	46.0	55.7	2/17/23 18:00	46.4	62.8	2/21/23 13:15	45.8	58.9
2/13/23 12:00	46.2	55.9	2/17/23 20:00	46.5	61.5	2/21/23 14:15	45.9	57.8
2/13/23 13:00	46.0	56.9	2/17/23 22:00	46.3	60.5	2/21/23 15:15	46.0	57.2
2/13/23 14:00	45.8	57.0	2/18/23 0:00	46.5	58.7	2/21/23 17:15	46.0	56.5
2/13/23 15:00	45.7	56.0	2/18/23 2:00	45.8	57.5	2/21/23 19:15	46.2	55.3
2/13/23 17:00	45.6	55.3	2/18/23 5:00	46.2	56.5	2/21/23 21:15	46.3	55.1
2/13/23 19:00	45.6	55.7	2/18/23 8:00	46.0	55.8	2/21/23 23:15	46.4	55.2
2/13/23 21:00	45.9	55.2	2/18/23 9:00	46.0	55.9	2/22/23 0:15	46.5	55.9

Table A.5 (cont.)

Cycle 1			Cycle 2			Cycle 3		
Time	**Temperature (°C)**	**Humidity (%)**	**Time**	**Temperature (°C)**	**Humidity (%)**	**Time**	**Temperature (°C)**	**Humidity (%)**
2/13/23 22:00	46.1	55.2				2/22/23 2:15	46.0	55.6
2/14/23 0:00	46.0	55.2						
2/14/23 1:00	46.9	56.5						
2/14/23 3:00	46.8	56.3						
2/14/23 5:00	46.0	56.2						
2/14/23 8:00	46.9	55.5						

Table A.6: Test condition – 10mm specimen, 55 ± 2°C, 40 ± 5% RH

Cycle 1			Cycle 2			Cycle 3		
Time	**Temperature (°C)**	**Humidity (%)**	**Time**	**Temperature (°C)**	**Humidity (%)**	**Time**	**Temperature (°C)**	**Humidity (%)**
3/1/23 9:00	53.6	37.2	3/6/23 6:45	56.3	40.9	3/8/23 13:00	57.0	44.0
3/1/23 9:15	54.6	38.1	3/6/23 7:45	53.7	41.4	3/8/23 14:00	55.0	44.1
3/1/23 9:30	54.1	42.6	3/6/23 8:45	53.7	40.2	3/8/23 14:30	55.5	44.6
3/1/23 9:45	55.3	44.4	3/6/23 9:45	53.2	43.9	3/8/23 15:00	55.7	44.4
3/1/23 10:00	55.8	43.1	3/6/23 10:15	53.3	44.5	3/8/23 15:30	56.3	45.0
3/1/23 10:15	56.4	42.5	3/6/23 11:00	53.9	43.3	3/8/23 16:00	55.2	43.2
3/1/23 10:45	56.6	44.3	3/6/23 12:00	54.6	40.1	3/8/23 16:30	55.1	42.5
3/1/23 11:15	55.8	44.8	3/6/23 12:30	55.0	44.4	3/8/23 17:00	56.1	37.3
3/1/23 11:45	55.5	44.1	3/6/23 13:30	55.8	44.0	3/8/23 17:30	56.5	40.8
3/1/23 12:15	55.2	43.7	3/6/23 15:00	55.9	42.1	3/8/23 19:30	55.5	40.5
3/1/23 12:45	55.2	44.5	3/6/23 16:15	55.9	42.1	3/8/23 21:30	55.5	40.1
3/1/23 13:15	54.9	44.3	3/6/23 16:45	55.5	40.0	3/8/23 22:30	56.8	37.2
3/1/23 13:45	54.9	44.2	3/6/23 18:00	55.7	40.7	3/8/23 23:30	56.2	39.4
3/1/23 14:15	55.2	44.9	3/6/23 19:00	55.7	40.7	3/9/23 0:15	56.0	42.6
3/1/23 15:15	55.8	44.4	3/6/23 20:00	55.6	43.3	3/9/23 0:45	56.3	41.6
3/1/23 16:15	56.0	43.4	3/6/23 21:00	55.9	41.3	3/9/23 3:00	56.6	41.4
3/1/23 17:15	56.0	43.2	3/6/23 22:00	56.1	40.2	3/9/23 4:30	56.4	41.3
3/1/23 18:15	55.9	43.7	3/6/23 23:00	55.6	40.0	3/9/23 6:30	56.2	39.3
3/1/23 19:15	56.3	43.0	3/7/23 0:00	54.5	39.9	3/9/23 7:30	55.6	37.9

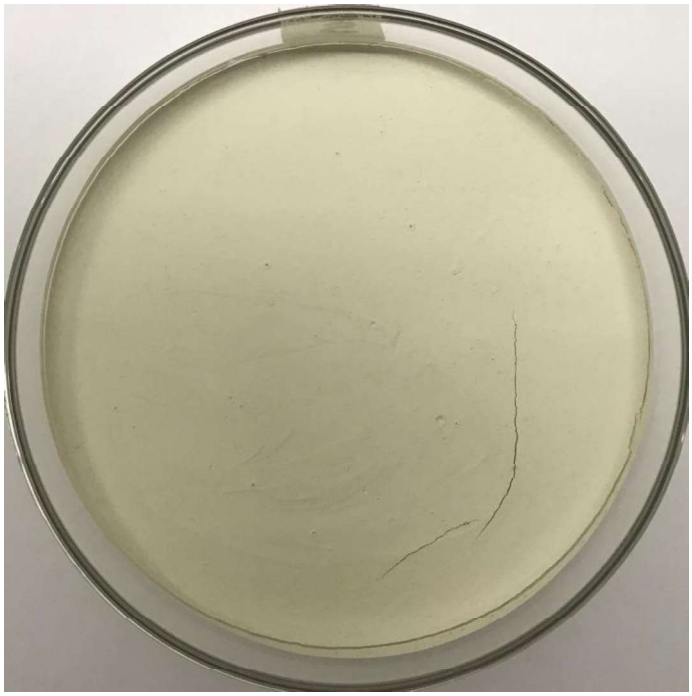
Table A.6 (cont.)

Cycle 1			Cycle 2			Cycle 3		
Time	**Temperature (°C)**	**Humidity (%)**	**Time**	**Temperature (°C)**	**Humidity (%)**	**Time**	**Temperature (°C)**	**Humidity (%)**
3/1/23 20:15	55.8	43.4	3/7/23 3:00	54.1	38.1	3/9/23 8:00	56.7	40.5
3/1/23 21:15	55.0	42.1	3/7/23 4:30	55.1	36.1	3/9/23 8:30	56.8	38.1
3/1/23 22:15	54.4	42.8	3/7/23 6:00	55.1	36.1	3/9/23 9:30	56.8	36.8
3/1/23 23:15	54.1	43.9	3/7/23 7:00	56.0	37.1	3/9/23 10:00	57.0	35.4
3/2/23 0:15	54.0	41.4	3/7/23 7:30	55.9	36.6	3/9/23 10:30	56.2	35.7
3/2/23 1:15	54.0	39.7	3/7/23 10:30	56.5	35.3	3/9/23 11:30	54.2	37.1
3/2/23 3:15	54.5	42.3	3/7/23 11:00	56.5	35.3			
3/2/23 6:15	55.1	41.3	3/7/23 11:30	56.6	35.9			
3/2/23 7:15	55.1	37.1						
3/2/23 9:15	55.1	42.6						
3/2/23 11:15	55.5	38.8						
3/2/23 12:15	55.8	37.9						
3/2/23 13:15	57.0	36.9						
3/2/23 15:30	57.0	35.2						
3/2/23 16:30	56.9	35.3						
3/2/23 17:30	56.6	35.7						

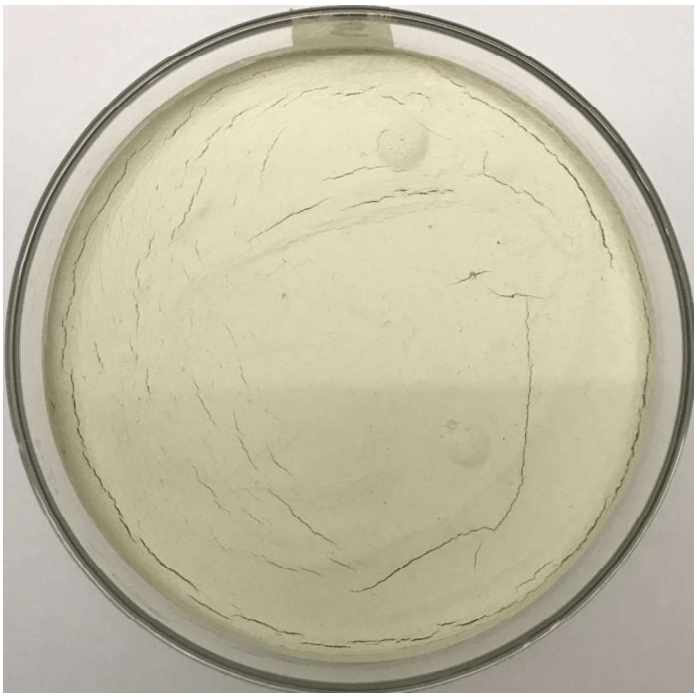
Appendix B

Soil Desiccation Original Images from The Experiment

This appendix contains examples of the original, unprocessed crack images from the experiment. Each image caption includes the test condition and specimen thickness. Only images captured at the end of the final wetting and drying cycle for each test condition are shown.



(a) W-D Cycle 1



(b) W-D Cycle 2

(c) W-D Cycle 3

(d) W-D Cycle 4

Figure B.1: 5mm specimen with test condition 45 ± 2°C, 40 ± 5% RH

(a) W-D Cycle 1

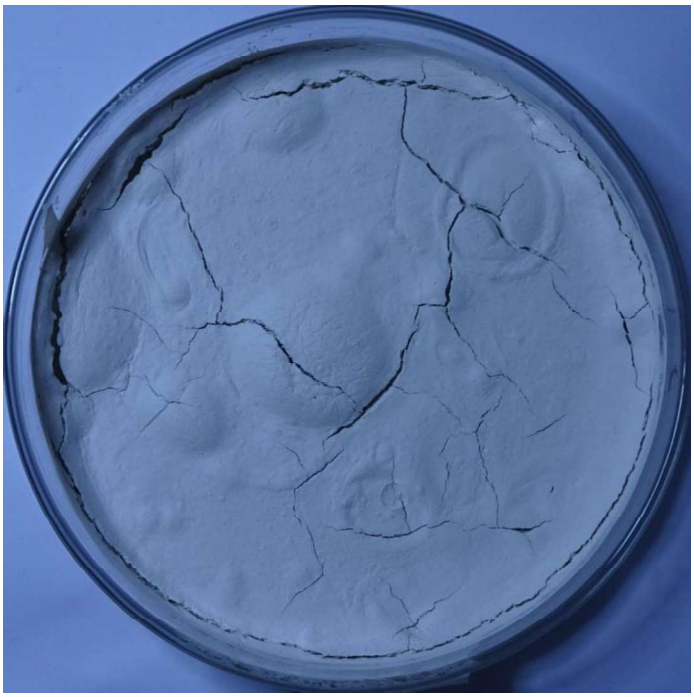
(b) W-D Cycle 2

(c) W-D Cycle 3

Figure B.2: 5mm specimen A with test condition 45 ± 2°C, 60 ± 5% RH

(a) W-D Cycle 1

(b) W-D Cycle 2



(c) W-D Cycle 3

Figure B.3: 5mm specimen B with test condition 45 ± 2°C, 60 ± 5% RH

(a) W-D Cycle 1

(b) W-D Cycle 2

(c) W-D Cycle 3

(d) W-D Cycle 4

Figure B.4: 10mm specimen with test condition 45 ± 2°C, 40 ± 5% RH

(a) W-D Cycle 1

(b) W-D Cycle 2

(c) W-D Cycle 3

Figure B.5: 10mm specimen A with test condition 45 ± 2°C, 60 ± 5% RH

(a) W-D Cycle 1

(b) W-D Cycle 2

(c) W-D Cycle 3

Figure B.6: 10mm specimen B with test condition 45 ± 2°C, 60 ± 5% RH

(a) W-D Cycle 1

(b) W-D Cycle 2

(c) W-D Cycle 3

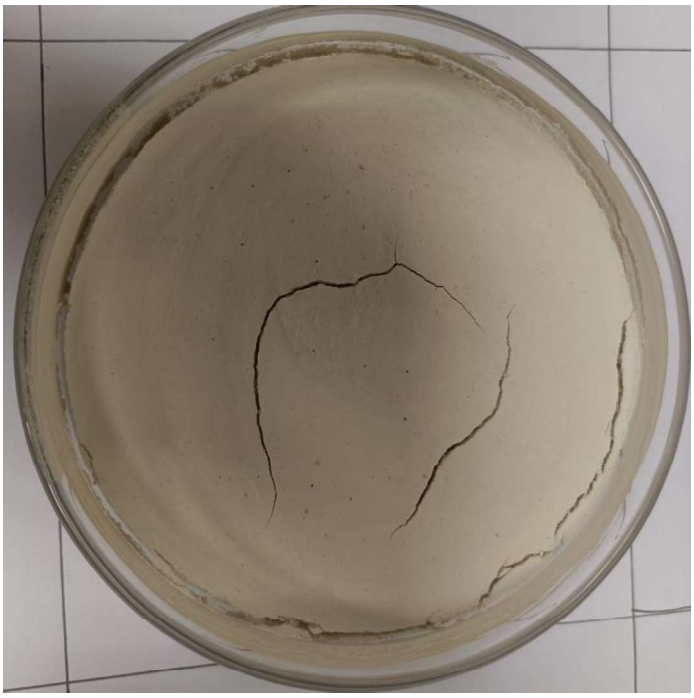
Figure B.7: 10mm specimen A with test condition 55 ± 2°C, 40 ± 5% RH

(a) W-D Cycle 1

(b) W-D Cycle 2

(c) W-D Cycle 3

Figure B.8: 10mm specimen B with test condition 55 ± 2°C, 40 ± 5% RH

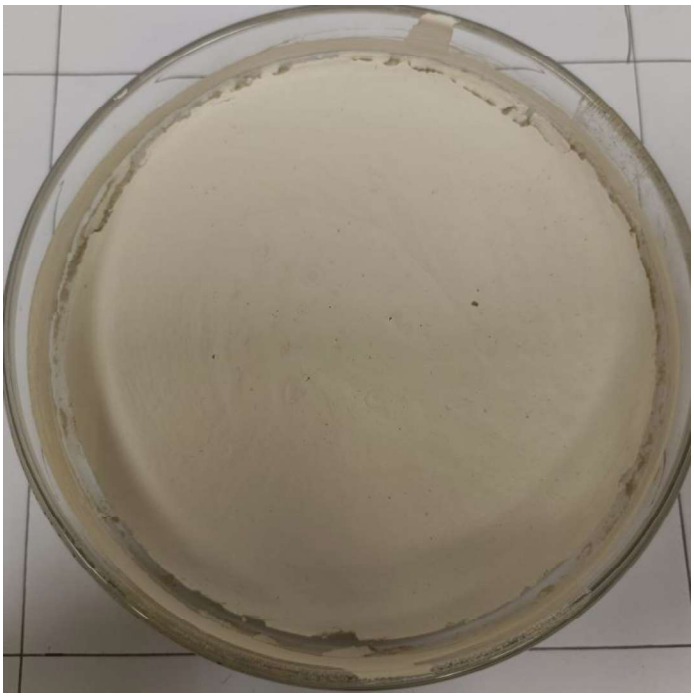


(a) W-D Cycle 1

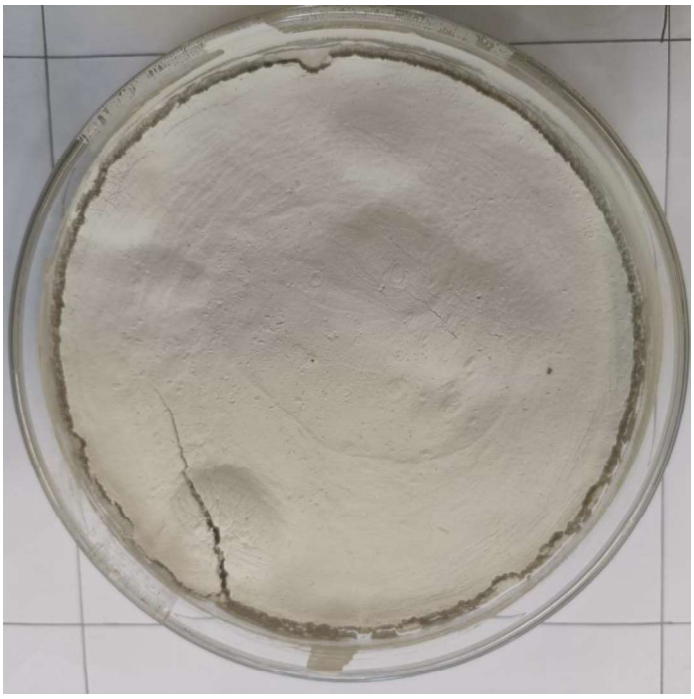
(b) W-D Cycle 2

(c) W-D Cycle 3

Figure B.9: 10mm specimen A with test condition 105°C, 0 % RH



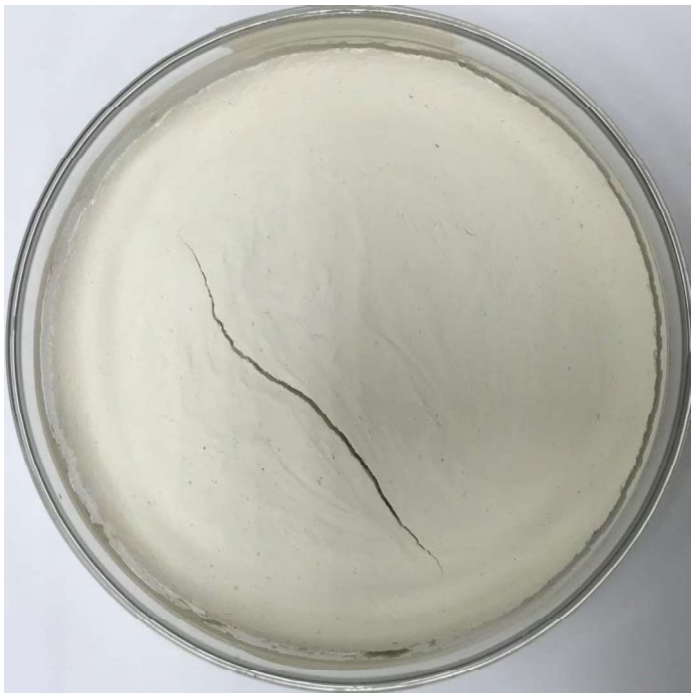
(a) W-D Cycle 1



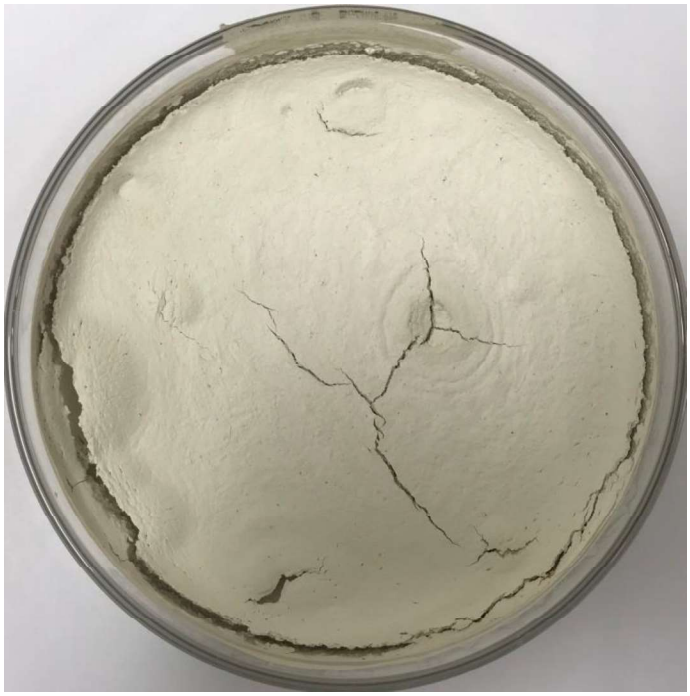
(b) W-D Cycle 2

(c) W-D Cycle 3

Figure B.10: 10mm specimen B with test condition 105°C, 0 % RH



(a) W-D Cycle 1



(b) W-D Cycle 2

(c) W-D Cycle 3

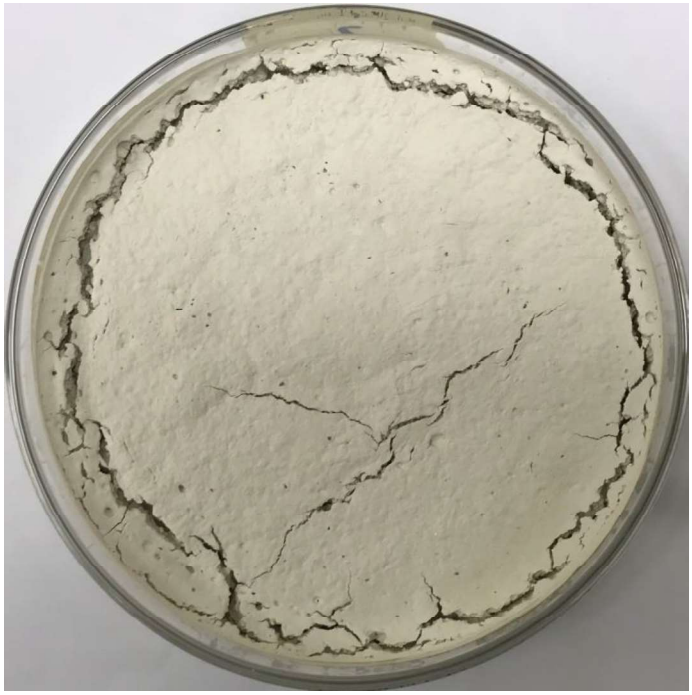
(d) W-D Cycle 4

Figure B.11: 10mm specimen C with test condition 105°C, 0 % RH

(a) W-D Cycle 1

(b) W-D Cycle 2

(c) W-D Cycle 3



(d) W-D Cycle 4

Figure B.12: 10mm specimen D with test condition 105°C, 0 % RH

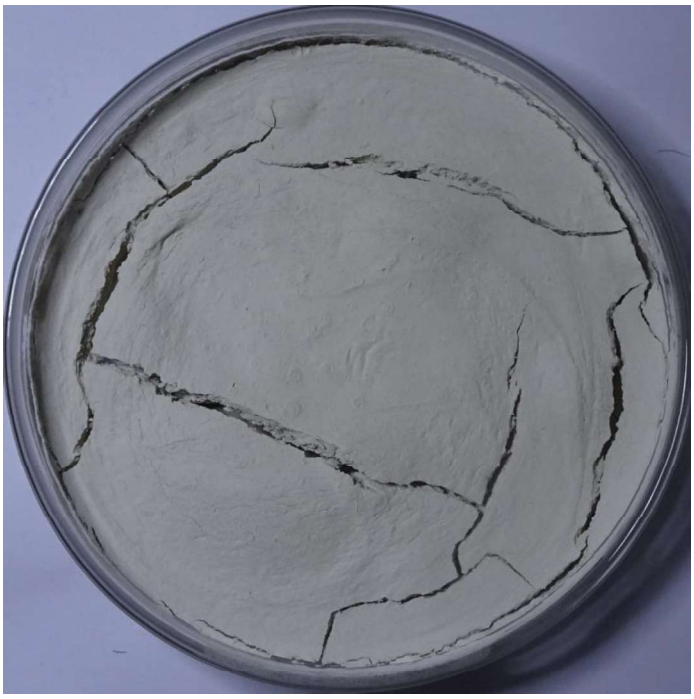
(a) W-D Cycle 1

(b) W-D Cycle 2

(c) W-D Cycle 3

Figure B.13: 15mm specimen A with test condition 45 ± 2°C, 60 ± 5% RH

(a) W-D Cycle 1



(b) W-D Cycle 2

(c) W-D Cycle 3

Figure B.14: 15mm specimen B with test condition 45 ± 2°C, 60 ± 5% RH